

(12) PATENT		(11) Application No. AU 199911271 B2	
(19) AUSTRALIAN PATENT OFFICE		(10) Patent No. 727677	
(54)	Title A method for smoothing jagged edges in digital images		
(51) ⁶	International Patent Classification(s) G06T 005/00 G06K 009/44		
(21)	Application No:	199911271	(22) Application Date: 1999.01.11
(30)	Priority Data		
(31)	Number	(32) Date	(33) Country
	PPI284	1998.01.12	AU
(43)	Publication Date: 1999.08.19		
(43)	Publication Journal Date: 1999.08.19		
(44)	Accepted Journal Date: 2000.12.21		
(71)	Applicant(s) Canon Kabushiki Kaisha		
(72)	Inventor(s) Delphine Anh Dao Le		
(74)	Agent/Attorney SPRUSON and FERGUSON, GPO Box 3898, SYDNEY NSW 2001		
(56)	Related Art US 5396584 US 5657430 EP 809207		

A Method for Smoothing Jagged Edges in Digital Images

Abstract

A method and apparatus for detection and smoothing of jagged edges in a graphics display is described. The edge detection and smoothing is done using only a pixel-map representation of the pixel colors of the images input. Edges are detected based upon the color representation of certain groups of pixels. Smoothing of the edges is accomplished by adjusting the color of certain pixels in the region of a detected edge. Edges are detected by successively scanning either sets of two rows or sets of two columns of pixels. Edge detection occurs during the row or column scans by discerning certain sequences of color changes between pairs of adjacent pixels. Jagged edges are detected and smoothed regardless of the type of color representation used in the pixel-map. The pixel-map may represent, for instance, the color of each pixel as a combination of shades of red, green or blue, a combination of varying degrees of blue, brightness and saturation, or some other combination.



S & F Ref: 441473

AUSTRALIA

PATENTS ACT 1990

COMPLETE SPECIFICATION

FOR A STANDARD PATENT

ORIGINAL

Name and Address
of Applicant:

Canon Kabushiki Kaisha
30-2, Shinomaruko 3-chome
Ohta-ku
Tokyo 146
JAPAN

Actual Inventor(s): Delphine Anh Dao Le

Address for Service: Spruson & Ferguson, Patent Attorneys
Level 33 St Martins Tower, 31 Market Street
Sydney, New South Wales, 2000, Australia

Invention Title: A Method for Smoothing Jagged Edges in Digital Images

ASSOCIATED PROVISIONAL APPLICATION DETAILS

[31] Application No(s)	[33] Country	[32] Application Date
PP1284	AU	12 January 1998

The following statement is a full description of this invention,
including the best method of performing it known to me/us:-

5815

A METHOD FOR SMOOTHING JAGGED EDGES IN DIGITAL IMAGES

The present invention relates to a method and apparatus for the directional smoothing of jagged edges in grey scale or colour images, in particular for text or graphics or any thin features.

BACKGROUND

Images represented in a pixel-based display often suffer from a staircase effect apparent in oblique lines and curves. This can adversely affect text readability, and especially the readability of small fonts applied to text. When an image is displayed at a higher resolution than the resolution with which it was generated, irregular edges can appear more pronounced.

For example, a ferro-electric liquid crystal display (FLCD) panel is capable of displaying text and natural image data at a standard screen resolution of 90 pixels per inch with a colour range of 4096 distinct colours. The panel is also capable of doubling the resolution with a consequently more limited colour range. Thus, it is possible to take advantage of the increased resolution to make text easier to read.

A number of conventional text enhancement systems rely on identifying predetermined patterns or features before modifying an image. For example, they modify the bitmap or pixel map depending on the positional relationships of background and foreground (text) pixels. This approach lacks generality and may not be able to handle exotic characters or particular graphics. Such a conventional technique assumes that the input data contains only text information. Alternatively, such a technique assumes that zone segmentation has previously discriminated between text data and natural image information. Moreover, the colour of the background and the foreground has to be known. Thus, such techniques are disadvantageous in that it is difficult to apply such techniques to complex images (eg., multiple text and background colours, or text on a natural background), or when no a priori colour information is available.

Alternatively, other conventional image processing methods like low-pass filtering can be used to smooth images. Such techniques do not depend on image content or colour information of an image. However, these techniques are not suitable for smoothing text and graphics, because they tend to blur high-contrast edges. The smoothing process should only be applied to jagged edges and preserve sharp discontinuities. Such conventional techniques disadvantageously smooth an image across edges in the image.

Still further, conventional antialiasing techniques are often based on high-level information regarding image content such as geometrical shapes or character fonts. Consequently, such techniques are limited to applications where the features in a

FLCD20 385682 CFPunkn

lo:cs:rel/codnew/flcd20/385682.doc

particular image are known and are disadvantageous in that image content is not always known and the image would have to be preprocessed to be reinterpreted at a higher level of abstraction.

5 A need clearly exists for an edge smoothing technique that overcomes one or more of the disadvantages of such conventional techniques. A general colour-independent, smoothing method is needed that can detect local edge information and avoid blurring sharp discontinuities. It is further desirable that such a technique preferably be capable of being applied to any kind of input data (ie. text, graphics or natural images), without prior segmentation to detect regions of interest or user
10 intervention. A need exists for a technique that can smooth jagged lines, improve overall text appearance and readability, without blurring images or altering textures such as halftoned images.

SUMMARY

15 In accordance with a first aspect of the invention, there is provided a method of smoothing jagged edges in graphical data. The method comprises steps of detecting one or more edges of a selected pixel in the graphical data dependent upon intensities of the selected pixel and another pixel surrounding a respective site of the one or more edges and predetermined gradients of at least the selected pixel and the
20 respective site surrounding pixel, and applying an adapted convolution mask to the selected pixel and a predetermined neighbourhood of pixels containing the selected pixel, wherein coefficient values of the convolution mask are dependent upon the one or more detected edges.

25 Preferably, the detecting step is dependent upon a difference between the intensities of the selected pixel and the site surrounding pixel satisfying an intensity threshold. It may also be dependent upon an absolute value of the difference exceeding the intensity threshold.

30 The detecting step may be dependent upon the predetermined gradients of the selected pixel and the site surrounding pixel satisfying a gradient threshold. The predetermined gradients are either vertical or horizontal dependent upon the edge site being either horizontal or vertical, respectively. The predetermined gradients are either greater than the gradient threshold, or are less than the gradient threshold.

35 Preferably, the detecting step is dependent upon predetermined gradients of diagonal pixels in the neighbourhood satisfying a gradient threshold. The predetermined gradients of the selected pixel and the site surrounding pixel may be greater than the gradient threshold, and the predetermined gradients of the diagonal pixels may be less than the gradient threshold. Alternatively, the predetermined gradients of the selected pixel and the site surrounding pixel may be less than the

gradient threshold, and the predetermined gradients of the diagonal pixels may be greater than the gradient threshold.

Preferably, if a single edge is detected, one or more coefficient values of a portion of the convolution mask corresponding to an edge containing the site
5 surrounding pixel is set to a predetermined value. The predetermined value may be zero.

Preferably, if a double edge is detected, one or more coefficient values of each portion of the convolution mask corresponding to a respective edge containing one of the two site surrounding pixels are set to a predetermined value. The
10 predetermined value may be zero.

Optionally, the method may comprise the step of, for a detected double edge, determining whether the selected pixel is a thickening pixel dependent upon a horizontal gradient and a vertical gradient of a pixel at the junction of the two edges satisfy respective gradient thresholds. One or more coefficient values of a portion of
15 the convolution mask corresponding to each edge opposite to the detected double edges may be set to a predetermined value.

Preferably, the detecting step is carried out for each colour component of the selected pixel, and the one or more detected edges are dependent upon the largest intensity and gradient values of the colour components.

In accordance with a second aspect of the invention, there is provided a method of smoothing jagged edges in graphical data. The method comprises the steps of: determining one or more edges if the intensities of a target pixel and a neighbouring pixel, both abutting a respective edge site, are substantially different, predetermined gradients dependent upon the respective edge site are an absolute local
20 maxima, and predetermined gradients of diagonal pixels are not an absolute local maxima of the same sign as that of the neighbouring pixel; and modifying the target pixel using a filter having coefficient values dependent upon the one or more determined edges.

In accordance with a third aspect of the invention, there is disclosed a
30 method of smoothing jagged edges in graphical data, the method comprising the steps of: determining one or more edges if the intensities of a target pixel and a neighbouring pixel, both abutting a respective edge site, are substantially different, and predetermined gradients dependent upon the respective edge site are an absolute local maxima; and modifying the target pixel using a filter having coefficient values
35 dependent upon the one or more determined edges.

In accordance with a fourth aspect of the invention, there is provided a method of smoothing jagged edges in graphical data, the method comprising the steps of: (a) detecting a plurality of zones of the graphical data dependent upon the values

of a plurality of pixels contained in each zone, the zones each containing a minority of either bright or dark pixels or containing an equal number of bright and dark pixels; and (b) for each zone, filtering pixels of the zone, the zone filtering comprising the steps of:

- 5 if the bright and dark pixels are substantially equally present, applying antialiasing to the zone; if the dark pixels are in a minority in the zone, applying thickening to the zone; and if the bright pixels are in a minority in the zone, applying thickening to the zone with relevant gradients negated.

- 10 In accordance with a fifth aspect of the invention, there is provided an apparatus for smoothing jagged edges in graphical data, the apparatus comprising: means for detecting one or more edges of a selected pixel in the graphical data dependent upon intensities of the selected pixel and another pixel surrounding a respective site of the one or more edges and predetermined gradients of at least the
15 selected pixel and the respective site surrounding pixel; and filtering means for applying an adapted convolution mask to the selected pixel and a predetermined neighbourhood of pixels containing the selected pixel, wherein coefficient values of the convolution mask are dependent upon the one or more detected edges.

- 20 In accordance with a sixth aspect of the invention, there is provided an apparatus for smoothing jagged edges in graphical data, the apparatus comprising: means for determining one or more edges if the intensities of a target pixel and a neighbouring pixel, both abutting a respective edge site, are substantially different, predetermined gradients dependent upon the respective edge site are an absolute local maxima, and predetermined gradients of diagonal pixels are not an absolute local
25 maxima of the same sign as that of the neighbouring pixel; and filtering means for modifying the target pixel, the filtering means having coefficient values dependent upon the one or more determined edges.

- 30 In accordance with a seventh aspect of the invention, there is provided an apparatus for smoothing jagged edges in graphical data, the apparatus comprising: means for determining one or more edges if the intensities of a target pixel and a neighbouring pixel, both abutting a respective edge site, are substantially different, and predetermined gradients dependent upon the respective edge site are an absolute local maxima; and filter means for modifying the target pixel, the filter means having coefficient values dependent upon the one or more determined edges.

- 35 In accordance with an eighth aspect of the invention, there is provided an apparatus for smoothing jagged edges in graphical data, the apparatus comprising: means for detecting a plurality of zones of the graphical data dependent upon the values of a plurality of pixels contained in each zone, the zones each containing a

minority of either bright or dark pixels or containing an equal number of bright and dark pixels; and means for filtering pixels of each zone, the zone filtering means further comprising:

- 5 means for antialiasing the zone if the bright and dark pixels are substantially equally present; and means for thickening the zone if the dark pixels are in a minority in the zone or if the bright pixels are in a minority in the zone.

10 In accordance with a ninth aspect of the invention, there is provided a computer program product including a computer readable medium having recorded thereon a computer program for smoothing jagged edges in graphical data, the computer program product comprising: means for detecting one or more edges of a selected pixel in the graphical data dependent upon intensities of the selected pixel and another pixel surrounding a respective site of the one or more edges and predetermined gradients of at least the selected pixel and the respective site
15 surrounding pixel; and filtering means for applying an adapted convolution mask to the selected pixel and a predetermined neighbourhood of pixels containing the selected pixel, wherein coefficient values of the convolution mask are dependent upon the one or more detected edges.

20 In accordance with a tenth aspect of the invention, there is provided a computer program product including a computer readable medium having recorded thereon a computer program for smoothing jagged edges in graphical data, the computer program product comprising: means for determining one or more edges if the intensities of a target pixel and a neighbouring pixel, both abutting a respective edge site, are substantially different, predetermined gradients dependent upon the
25 respective edge site are an absolute local maxima, and predetermined gradients of diagonal pixels are not an absolute local maxima of the same sign as that of the neighbouring pixel; and filtering means for modifying the target pixel, the filtering means having coefficient values dependent upon the one or more determined edges.

30 In accordance with an eleventh aspect of the invention, there is provided a computer program product including a computer readable medium having recorded thereon a computer program for smoothing jagged edges in graphical data, the computer program product comprising: means for determining one or more edges if the intensities of a target pixel and a neighbouring pixel, both abutting a respective edge site, are substantially different, and predetermined gradients dependent upon the
35 respective edge site are an absolute local maxima; and filter means for modifying the target pixel, the filter means having coefficient values dependent upon the one or more determined edges.

In accordance with a twelfth aspect of the invention, there is provided a computer program product including a computer readable medium having recorded thereon a computer program for smoothing jagged edges in graphical data, the computer program product comprising: means for detecting a plurality of zones of the graphical data dependent upon the values of a plurality of pixels contained in each zone, the zones each containing a minority of either bright or dark pixels or containing an equal number of bright and dark pixels; an means for filtering pixels of each zone, the zone filtering means further comprising:

means for antialiasing the zone if the bright and dark pixels are substantially equally present; means for thickening the zone if the dark pixels are in a minority in the zone or if the bright pixels are in a minority in the zone.

BRIEF DESCRIPTION OF THE DRAWINGS

Embodiments of the invention are described hereinafter with reference to the drawings, in which:

Fig. 1 is a flow diagram illustrating data flow for an FLCD panel;

Fig. 2 is a block diagram of a general purpose computer with which the embodiments of the invention can be practiced;

Fig. 3 illustrates a processing window containing a centre pixel surrounded by four edge sites;

Fig. 4 illustrates a processing window, like that of Fig. 3, including the surrounding pixels that are considered for the edge detection at the eastern edge site;

Fig. 5 illustrates the naming convention of a smoothing filter;

Figs. 6A-6C are flow diagrams illustrating an edge detection process in accordance with a first embodiment of the invention;

Fig. 7 is a flow diagram illustrating a smoothing or filtering process in accordance with the first embodiment of the invention

Figs. 8A to 8C illustrate block diagrams of exemplary input data containing a diagonal line, edge detection results, and filtered output data produced in accordance with the first embodiment shown in Figs. 6A-6C and 7A;

Figs. 9A to 9D are flow diagrams of post-processing for subsequent halftoning of a smoothed image in accordance with the first embodiment;

Fig. 10A shows patterns of modified pixels which can be allowed;

Fig. 10B illustrates high contrast textures of a dithered image which are preserved by the smoothing method;

Fig. 11 is a flow diagram of the edge detection process, for use in combination with the processes of Figs. 6A and 6B, in accordance with the second embodiment;

Fig. 12 is a flow diagram illustrating the smoothing process in accordance with the second embodiment;

Figs. 13A to 13C illustrate the effect of the smoothing method described in the second embodiment on a diagonal line;

5 Figs. 14A and 14B illustrate the effect of smoothing on 'English' characters according to the second embodiment of the invention;

Figs. 15A to 15D illustrate the effect of smoothing on Japanese characters according to the second embodiment;

Fig. 16 shows a pixel thickening an oblique line and its neighbours;

10 Figs. 17A to 17D illustrate the colour detection process according to the embodiments of the invention;

Figs. 18A to 18E are block diagrams illustrating the detection of an exemplary vertical transition using the colour detection process of Figs. 17A-17D;

15 Figs. 19A to 19C illustrate the process of pixel classification in the colour detection process of Figs. 17A-17D;

Fig. 20 is a flow diagram illustrating a variation of the process of smoothing thickening candidate pixels in accordance with the second embodiment;

Fig. 21 is a flow diagram illustrating a combined antialiasing and thickening process in accordance with the third embodiment of the invention; and

20 Figs. 22A to 22C illustrate the process of comparing original and antialiased images in accordance with the third embodiment.

DETAILED DESCRIPTION

1. Overview

25 The present invention relates to an edge smoothing method and apparatus that preserves edges while smoothing jagged edges. This principally involves detecting edges in the input data, and modifying the smoothing filter dependent upon the edge detection results. The input data to this smoothing technique is a pixel-map representation of a pixel grid at a display resolution. Relevant data with jagged edges includes characters intended for use at a first resolution but which are used at a second, higher resolution. The embodiments of the invention have numerous applications, and are particularly useful in relation to image data having jagged edges such as text characters intended for display on a low-resolution display device, but which are displayed on a high-resolution display device (e.g., an FLCD panel having a display resolution that is twice the input resolution). The smoothing method assumes that the data have previously been increased to a higher resolution, or upsampled, without attempting to smooth sharp edges (e.g., by replicating the input pixels into four

identical subpixels). Using the edge detection and smoothing processes of the embodiments of the invention, the overall appearance and readability of the text is improved even if individual letters are not always improved.

In the following description, numerous specific details including particular resolutions, colour space formats, specific pixel-size processing windows, specific pixel-size convolution masks, etc. are described in detail to provide a more thorough description of the embodiments of the invention. However, it will be apparent to one skilled in the art that the present invention may be practiced without these specific details. In other instances, well-known features have not been described in detail so as not to unnecessarily obscure the present invention.

Fig. 1 illustrates an exemplary FLCDD data flow. The representation of images on an FLCDD typically involves a number of stages. For example, using an input image 100, upsampling is performed in step 102 which resolves the data from a lower resolution (eg. 90 dpi) to a higher resolution (eg. 180 dpi). The image data may then be smoothed in step 104, thereby reducing or eliminating any jagged edges. This step 104 is followed by steps 106 and 108 of halftoning and displaying the halftoned pixels, respectively.

The pixel-map comprises a numerical representation of a particular colour for each pixel location in a rectangular array. Any numerical representation of colour may be used as long as it is expressed as a sequence of one or more numbers. Preferably, the data information is supplied on a pixel-by-pixel basis in raster image order (ie. from left to right across a display device and from top to bottom of the display device in a non-interlaced fashion). Pixel locations at the pixel grid are represented by an array of row (i) and column (j) specifications. All the pixels are treated sequentially, and the output for a target pixel depends on the input data in a limited neighbourhood of this pixel.

The embodiments of the invention are preferably implemented using a conventional general-purpose (host) computer system, such as the computer system 200 shown in Fig. 2. Further, the processes of detecting and smoothing edges according to the embodiments of the invention can be embodied in an application program and are preferably implemented as software loaded into and carried out using the computer system 200. The software can be loaded into the computer system 200 from a computer readable medium including the storage devices described below, for example. The computer system 200 includes a computer module 200, one or more input devices such as a keyboard 228 and mouse 230, and one or more output devices such as a printer or other reproduction device 224 including plotters, laser printers, etc., and a computer display monitor 204.

A video interface/adaptor 206 connected to the video display monitor 204 provides video signals from the computer module 202 for display on the video display

FLCDD 20 385682 CFPurkin

file:///c:/cdnew/ffcd201385682.doc

monitor 204. A modulator-demodulator (Modem) transceiver device 222 may be used by the computer module 202 for communicating to and from a computer network, for example connectable via a telephone line or other functional medium. The modem 222 can be used to obtain access to the Internet and other network systems. Optionally, the computer system may also include a network adaptor (not shown) for connecting the computer to a computer network such as a local area network (LAN) or wide area network (WAN), for example. Other connection interfaces such as ISDN may be used also.

The computer module 202 typically includes at least one central processing unit (CPU) 212, a memory unit 216, for example formed from semiconductor random access memory (RAM) and read only memory (ROM), input/output (I/O) interfaces 206 and 208 including a video interface 206, and an I/O interface 214 for the keyboard 228, a mouse 230 and optionally a joystick (not shown). A storage device 210 is provided and typically includes a hard disk drive 210A and a floppy disk drive 210B. The storage device 210 may also comprise a magneto-optical disc drive, magnetic tape, DVD, etc. A CD-ROM drive 218 is typically provided as a non-volatile source of data.

The components 206 to 210 and 212 to 218 of the computer module 202 typically communicate via an interconnected bus 220 and in a manner which results in a conventional mode of operation of the computer system 200 known to those in the relevant art. Such a bus 220 typically comprises data, address, and control buses. However, the overall structure and individual components of the computer system 600 are conventional and are well known to persons skilled in the art.

Thus, the system 200 is simply provided for illustrative purposes and other configurations can be employed without departing from the scope and spirit of the invention. The computer system may be implemented using an IBM personal computer (PC) or compatible, one of the Macintosh (TM) family of PCs, Sun Sparcstation (TM), or the like. The foregoing are merely exemplary of the types of computers with which the embodiments of the invention may be practiced. Typically, the application program of the preferred embodiment is resident as software recorded on a hard disk drive 210A as the computer readable medium, and read and controlled using the processor 212. Intermediate storage of the program and pixel data and any data fetched from the network may be accomplished using the semiconductor memory 216, possibly in concert with the hard disk drive 210A. In some instances, the application program may be supplied to the user encoded on a CD-ROM 218 or a floppy disk 210B, or alternatively could be read by the user from the network via the modem device 222, for example. Still further, the software can also be loaded into the computer system 200 from other computer readable medium including magnetic tape, a ROM or integrated circuit, a magneto-optical disk, a radio or infra-red transmission

channel between the computer and another device, a computer readable card such as a PCMCIA card, and the Internet and Intranets including email transmissions and information recorded on websites and the like. The foregoing are merely exemplary of relevant computer readable mediums. Other computer readable mediums may be practiced without departing from the scope and spirit of the invention.

Embodiments of the invention are described in detail hereinafter. Again, such embodiments may preferably be implemented using the representative computer system 200.

10 2. Antialiasing Process of First Embodiment

The edge-preserving smoothing process according to a first embodiment comprises two component processes: edge detection and smoothing. The edge detection process determines the presence and direction of edges of target pixels of input image data. Once the results of the edge detection process are determined, the edge smoothing process applies a linear smoothing filtering process to the target pixels. The filtering process itself is modified dependent on any detected edges to preserve such detected edges in the image data. The edge detection and smoothing processes are described in greater detail hereinafter.

20 2.1. Edge Detection Process

As noted above, the edge detection process aims at determining the presence and direction of edges. The process first involves computing and thresholding relevant horizontal and vertical gradients of pixels. Then, edge information is collected for each target pixel by turning "ON" or "OFF" four surrounding edge sites of a target pixel being processed.

2.1.1. Gradient Computation and Thresholding

Gradient computation and thresholding is used to detect discontinuities in image data. The process of discontinuity detection relies on a first-order derivative method requiring the generation of gradients in two orthogonal (horizontal and vertical) directions. This can be done using any orthogonal differential gradient operator. For example, two 3x3 pixel convolution masks corresponding to the Sobel edge detector may be used. The Sobel gradient operator consists of the horizontal and vertical convolution kernels shown in Table 1.

TABLE 1

	-1	0	1
1/4	-2	0	2
	-1	0	1

Horizontal gradient operator

	-1	-2	-1
1/4	0	0	0
	1	2	1

Vertical gradient operator

Discontinuities are characterised by high gradient values and high intensity differences between neighbouring pixels. Thus, the edge detection process of the first embodiment defines two thresholds: one for the gradient, and another for the intensity.

10 The first threshold is determined to detect a local maximum of either a vertical or horizontal gradient, so that edges can be detected regardless of the contrast. A value T_g can be such a gradient threshold. The threshold T_g preferably corresponds to a predetermined percentage of the absolute maximum value of vertical and horizontal components of the gradient of all colour components in the neighbourhood of the target pixel. For example, T_g can be set to one third of the absolute maximum gradient value on a 3×3 pixel window centred on the pixel being currently processed (ie, the target pixel).

20 The second threshold T_f is used to detect significant intensity differences between pixels. It may also be defined as a predetermined percentage of the total intensity range, typically a small percentage (e.g., one percent) if the images are noise-free. The threshold T_f may be defined independently of the pixel location. For instance, a unique intensity threshold may be used for the whole image. The relevant gradient threshold, intensity difference threshold, and computed gradient values are used in the edge site determination process described hereinafter.

2.1.2. Edge Site Determination Process

For each pixel, the edge detection process of the first embodiment depends on the gradient and intensity values in a 3×3 pixel neighbourhood as shown in Fig. 3.

- 5 Each target pixel 302 in the centre of the neighbourhood is treated as being surrounded by four potential edge sites, labelled East 306, North 304, West 310 and South 308 according to the corresponding compass directions. The East and West edge sites 306, 310 correspond to vertical edges. In contrast, the North and South edge sites 304, 308 correspond to horizontal edges. The presence or absence of an edge determines whether
- 10 those sites are "ON" or "OFF".

A principal difficulty for the smoothing process is to distinguish between oblique lines or curves, which suffer from a staircase effect and need to be smoothed, and horizontal or vertical edges that are to be preserved without smoothing. According to the first embodiment of the invention, the edge sites are turned on only

15 if they belong to a true horizontal or vertical line (which is more than 2 pixel long) and not to a diagonal line.

A vertical edge site is turned on if the following conditions are met:

- 20 A) the intensities of the two pixels surrounding the edge site (i.e., the target pixel and either the relevant left or right pixel) are significantly different;
- B) the horizontal gradient values of those two surrounding pixels are an absolute local maxima; and
- C) the horizontal gradient values of the relevant diagonal pixels are not an absolute local maxima of the same sign as the surrounding pixels (i.e.,
- 25 the target pixel is not likely to belong to a diagonal line).

- For example, Fig. 4 is a block diagram of a group of pixels 300 indicating the relevant pixels used to determine whether an East edge site 404 is "ON" or "OFF". The group of pixels is centred on a target pixel 400. The target pixel 400 and the right pixel 402 surround the edge site 404. In the foregoing process for
- 30 determining if a relevant vertical (ie, East) edge site is on, the difference in intensities of the two pixels 400, 402 and the horizontal gradients of the two pixels 400, 402 are checked. Further, upper left pixel 401 and lower left pixel 403 are the relevant diagonal pixels (relative to the target pixel 400) considered for the East edge site 404. The horizontal gradient values of these pixels are also checked. Pixels of the group
- 35 300 are analogously used to determine the state of the other horizontal edge site (i.e., West).

Similarly, a horizontal edge site is turned on if the following conditions are met:

FLCD20 385682 CFHunkn

lo:\clar\lfdnew\lfd20\385682.doc

- A) the intensities of the two pixels surrounding the edge site (i.e., the target pixel and either the relevant top or bottom pixel) are significantly different;
- B) the vertical gradient values of those two surrounding pixels are an absolute local maxima; and
- C) the vertical gradient values of the relevant diagonal pixels are not an absolute local maxima of the same sign as the surrounding pixels.

Again, analogous pixels of the group 300 shown in Fig. 4 are used to determine the state of the horizontal edge site (i.e., North or South).

- A more precise statement of the rules for determining if edge sites are on is given in Table 2.

TABLE 2

- Rule 1: The EAST edge is "ON" if:
 - (i) $\text{abs}((F(i,j) - F(i,j+1))) > T_f$; AND
 - (ii) EITHER:
 - $(G_x(i,j) > T_g \text{ and } G_x(i,j+1) > T_g \text{ and } G_x(i-1,j-1) < T_g \text{ and } G_x(i+1,j-1) < T_g)$,
 - OR
 - $(G_x(i,j) < -T_g \text{ and } G_x(i,j+1) < -T_g \text{ and } G_x(i-1,j-1) > -T_g \text{ and } G_x(i+1,j-1) > -T_g)$.
- Rule 2: The WEST edge is "ON" if:
 - (i) $\text{abs}((F(i,j) - F(i,j-1))) > T_f$; AND
 - (ii) EITHER:
 - $(G_x(i,j) > T_g \text{ and } G_x(i,j-1) > T_g \text{ and } G_x(i-1,j+1) < T_g \text{ and } G_x(i+1,j+1) < T_g)$,
 - OR
 - $(G_x(i,j) < -T_g \text{ and } G_x(i,j-1) < -T_g \text{ and } G_x(i-1,j+1) > -T_g \text{ and } G_x(i+1,j+1) > -T_g)$.
- Rule 3: The NORTH edge is "ON" if:
 - (i) $\text{abs}((F(i,j) - F(i-1,j))) > T_f$; AND
 - (ii) EITHER:
 - $(G_y(i,j) > T_g \text{ and } G_y(i-1,j) > T_g \text{ and } G_y(i+1,j-1) < T_g \text{ and } G_y(i+1,j+1) < T_g)$,
 - OR
 - $(G_y(i,j) < -T_g \text{ and } G_y(i-1,j) < -T_g \text{ and } G_y(i+1,j-1) > -T_g \text{ and } G_y(i+1,j+1) > -T_g)$.

Rule 4: The SOUTH edge is "ON" if:

(i) $\text{abs}((F(i,j) - F(i,j+1)) > T_f)$ AND

(ii) EITHER:

- 5 $(G_y(i,j) > T_g \text{ and } G_y(i,j+1) > T_g \text{ and } G_y(i-1,j-1) < T_g \text{ and } G_y(i-1,j+1) < T_g)$,
 OR
 $(G_y(i,j) < -T_g \text{ and } G_y(i,j+1) < -T_g \text{ and } G_y(i-1,j-1) > -T_g \text{ and } G_y(i-1,j+1) > -T_g)$.

- 10 In Table 2, the parameters i and j are the row and column numbers, respectively, of
 the pixel being currently processed (i.e., the target pixel). The terms $F(i,j)$, $G_x(i,j)$
 and $G_y(i,j)$ represent the intensity, horizontal gradient and vertical gradient,
 respectively, at the target pixel (i,j). The term $\text{abs}(x)$ refers to the absolute value of
 x. The edge detection rules for the four edge sites shown in Table 2 amount to
 15 turning the edge site on if two relevant conditions regarding (i) intensity and (ii)
 gradients are satisfied.

For colour images, the edge detection process is performed by testing all
 colour components of the target pixel using the rules set forth in Table 2. This may be
 done sequentially. Preferably, the edge information is inferred from the colour
 component that has the largest contrast or the largest absolute gradient value. When
 20 detecting the colour component that has the largest contrast, a different weight can be
 given to each colour component. For example, in the case of RGB images, weights
 can be chosen that are proportional to the weights used for the computation of
 luminance (i.e., 0.299 for the red colour component, 0.587 for the green colour
 component, and 0.114 for the blue colour component of the target pixel).

2.1.3. Flow Diagrams of the Generalised Edge Detection Process

- 30 Figs. 6A to 6C provide a generalised flow diagram providing an overview of
 the process of detecting edges contained in image data in accordance with the first
 embodiment.

- In particular, Fig. 6A provides one possible implementation at a high level for
 35 detecting edges in digital image data comprising a plurality of pixels, as well as the
 smoothing or filtering step described hereinafter in another section. Fig. 6B is an
 intermediate level flow diagram of the process of edge detection for a given target pixel.
 Finally, Fig. 6C is a flow diagram illustrating the process of edge site determination for

a given target pixel, where the steps are illustrated as generalised representations of the edge site determination rules stated in Table 2.

With regard to Fig. 6A, processing commences in step 600. In step 602, one of the pixels of the image data to be processed is selected as the first target pixel. In decision block 604, a check is made to determine if all pixels of the image data including the selected target pixel have been processed. If decision block 604 returns true (yes) indicating that all pixels have been processed, processing terminates at step 606.

If decision block 604 returns false (no) indicating that the selected target pixel has not yet been processed, processing continues at decision block 608. In decision block 608, a check is made to determine if edge detection has been done for the target pixel. If decision block 608 returns true (yes), processing continues at step 610. In step 610, smoothing or filtering is applied to the image data, if necessary, dependent upon the results of the edge detection process. This step is described in greater detail hereinafter with reference to Fig. 7. In step 612, the next target pixel is selected, and processing then continues at the decision block 604.

Otherwise if decision block 608 returns false (no), processing continues at step 614. In step 614, one of the colour components of the target pixel is selected for processing. While the first embodiment is described in relation to a multi-component colour value for each pixel, the present invention is equally applicable to grey scale images with minor changes to step 614 and decision block 616. In decision block 616, a check is made to determine if the gradient has been computed for each color component. If decision block 616 returns false (no), processing continues at step 620. In step 620, the gradient computation process is carried out using the currently selected colour component of the selected target pixel. In step 622, the next remaining color component, if any, is selected and processing continues at decision block 616.

If decision block 616 returns true (yes), processing continues at step 618. In step 618, edge data from 1 or more color components is determined using predetermined criteria. Processing then continues at decision block 608.

Fig. 6B is a flow diagram illustrating further details of the steps carried out to implement the edge detecting step 618 of Fig. 6A. In step 630, processing commences. In step 632, the colour component with the biggest contrast is selected, and the gradient thresholds are also determined, if necessary. Processing continues at step 634. In step 634, the East edge site is checked. In step 636, the West edge site is checked. In step 638, the North edge site is checked. In step 640, the South edge site is checked. The edge data determined from step 632 to 640 for the selected colour component of the target pixel is returned in step 642 to colouring processing. While particular ordering of

steps is shown in Fig. 6B, changes can be made to the ordering and nature of the steps without departing from the scope and spirit of the invention.

- Fig. 6C is a flow diagram of the generalised steps carried out to implement each of steps 634 to 640 in Fig. 6B so as to implement the edge site determination rule of Table 2. Processing commences in step 660. In step 662, a check is made to determine if a difference between the intensity values of the target pixel and the relevant other surrounding pixel (i.e., right, top, left or bottom pixel for the East, North, West or South edge site, respectively) is greater than the predetermined intensity threshold. Decision block 662 implements sub-rule (i) of each of the rules of Table 2.
- When decision block 660 returns false (no) indicating that the absolute value of the intensity different is not greater than the threshold intensity, processing continues at step 668. In step 668, the respective edge of the target pixel for the selective colour component is set to "OFF". Processing then continues at step 670 and the edge data is returned to the calling procedure.
- If decision block 662 returns true (yes), processing continues at decision block 664. In decision block 664, a check made to determine if the relevant gradients (i.e., the horizontal or vertical gradient) of the target pixel and the respective other surrounding pixel (i.e., right, top, left or bottom pixel) is either greater than the gradient threshold or less than the negative value of the gradient threshold. This check implements the first portion of the two alternative substeps of sub-rule (ii) if Table 2. If decision block 664 returns false (no), processing continues at step 668 as described hereinbefore. Otherwise, if decision block 664 returns true (yes), processing continues at decision block 666. In decision block 666, a check is made to determine if the relevant gradients (i.e., the horizontal or vertical gradient) of the two diagonal pixels are less than the threshold gradient or greater than the negative value of the threshold gradient, respectively. Decision block 666 implements the second substep of sub-rule (ii) of the rules in Table 2. If decision block 672 returns false (no), processing continues at step 668. Otherwise, if decision block 666 returns true (yes), processing continues at step 672. In step 672, the respective edge (i.e., East, North, West or South) of the target pixel is set as "ON". Processing then continues at step 670. While the flow diagrams of Figs. 6A to 6C illustrate specific steps and ordering of steps, it will be apparent to one skilled in the art that modifications and/or changes to the particular steps and/or ordering of steps can be made to the edge detection process of the first embodiment without departing from the scope and spirit of the invention.
- In the foregoing manner, horizontal or vertical lines or curves can be determined. As is described hereinafter, the presence of "ON" edge sites is used to modify smoothing or filtering processes so as to preserve specified edges.

2.2. Smoothing Process

The edge smoothing process of the first embodiment involves applying a linear smoothing filter to each colour component resulting in all the colour components being smoothed identically. To preserve the detected edges, the convolution kernel of the filter is modified dependent upon the outcome of the edge detection for a target pixel. In particular, the filter values corresponding to the pixels surrounding the detected edge are set to zero.

In the smoothing process, any linear smoothing filter may be used. The modifications to be made to the filter are described with reference to a convolution kernel 500 that is preferably 3x3 pixels in size, as shown in Fig. 5. The three kernel values in the first row, from the first column to the third column, are referred to as V_a , V_b and V_c , respectively. The three kernel values in the second row, from the first column to the third column, are V_d , V_e and V_f , respectively. The three values in the third row, from the first to the third column, are V_g , V_h and V_i , respectively. These values are normalised by a normalisation factor of $1/N$, where N is the sum of all coefficients. For example, a 4-neighbour mean filter may be used. The classical formulation for this particular convolution kernel is given in Table 3.

TABLE 3

	0	1	0
1/4	1	0	1
	0	1	0

Example of smoothing filter

In the convolution kernel of Table 3, the kernel values $V_b = V_d = V_f = V_h = 1$, the kernel values $V_a = V_c = V_e = V_g = V_i = 0$, and $N = 4$.

In the following description, the smoothing process of the first embodiment is described in relation to three possible forms of this processing, that is full smoothing, no smoothing, and modified smoothing.

2.2.2. Full Smoothing

If the four edge sites of a target pixel are "OFF" and either the horizontal or the vertical gradient is an absolute local minima, the target pixel is likely to belong to

an oblique line. Therefore, unmodified or full smoothing filter is applied to the target pixel.

2.2.3. No Smoothing

5

If the four edge sites are "OFF" and neither the horizontal nor the vertical gradient is an absolute local minima, there is no edge around the target pixel. Therefore, smoothing is not performed on the target pixel. The output value for the target pixel is equal to its input value. Thus, only pixels in the region of an edge are affected by the smoothing procedure. Other pixels are left unchanged to avoid unnecessary blurring of other parts of the image.

10

2.2.3. Modified Smoothing

15

If a single or multiple edge of a target pixel has been detected (i.e., one or more edge sites is "ON"), a smoothing filter is applied, where an expression for the smoothing filter depends on the site that is turned on. In the modified smoothing filter, all the filter values remain identical to the classical filter, except for the ones that are next to the detected edge. Those values are turned to zero to preserve the discontinuity at the edge, and the normalisation factor of the filter is adjusted accordingly.

20

2.2.4. Smoothing Rules

25

Smoothing or filtering of target pixels in accordance with the first embodiment involves applying full, no and modified smoothing to a target pixel (i,j) dependent upon the edge detection results. More precise statements of these smoothing rules for modifying the convolution kernel are set forth in Table 4.

30

TABLE 4

Rule 1: If the four edge sites are "OFF":

if $(\text{abs}(\text{Gx}(i,j)) > \text{Tg} \text{ and } \text{abs}(\text{Gy}(i,j)) > \text{Tg})$

$\text{Vb} = \text{Vd} = \text{Vf} = \text{Vh} = 1, \text{Va} = \text{Vc} = \text{Ve} = \text{Vg} = \text{Vi} = 0, \text{ and } \text{N} = 4$

35

else

$\text{Vb} = \text{Vd} = \text{Vf} = \text{Vh} = \text{Va} = \text{Vc} = \text{Ve} = \text{Vg} = \text{Vi} = 0.$

Rule 2: If one of the four edge sites is "ON", the filter values are the same as in Rule 1 except for:

5 EAST: $V_c = V_f = V_i = 0$ and $N = V_a + V_b + V_d + V_e + V_g + V_h$
 NORTH: $V_a = V_b = V_c = 0$ and $N = V_d + V_e + V_f + V_g + V_h + V_i$
 WEST: $V_a = V_d = V_g = 0$ and $N = V_b + V_c + V_e + V_f + V_h + V_i$
 SOUTH: $V_g = V_h = V_i = 0$ and $N = V_a + V_b + V_c + V_d + V_e + V_f$

Rule 3: If two edge sites are "ON", all the filter values are the same as in Rule 1 except for:

10 NORTH-EAST: $V_a = V_b = V_c = V_f = V_i = 0$ and $N = V_d + V_e + V_g + V_h$
 NORTH-WEST: $V_a = V_b = V_c = V_d = V_g = 0$ and $N = V_e + V_f + V_h + V_i$
 SOUTH-EAST: $V_c = V_f = V_g = V_h = V_i = 0$ and $N = V_a + V_b + V_d + V_e$
 SOUTH-WEST: $V_a = V_d = V_g = V_h = V_i = 0$ and $N = V_b + V_c + V_e + V_f$
 15 NORTH-SOUTH: $V_a = V_b = V_c = V_g = V_h = V_i = 0$ and $N = V_d + V_e + V_f$
 EAST-WEST: $V_a = V_c = V_d = V_f = V_g = V_i = 0$ and $N = V_b + V_e + V_h$

The rules can be generalised for the cases where more than 2 (i.e., 3 or 4) edge sites are on.

20 Thus, either full smoothing or no smoothing is applied in Rule 1 of Table 4. With the chosen filter, Rule 2 amounts to performing a 3-neighbour interpolation. Rule 3 amounts to performing a 2-neighbour interpolation. Table 5 illustrates the convolution masks for the specific smoothing filter of Table 3 implemented in accordance with the smoothing rules of Table 4.

TABLE 5

Single edges:

1/3

0	1	0
1	0	0
0	1	0

EAST

1/3

0	0	0
1	0	1
0	1	0

NORTH

$$\frac{1}{3} \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

WEST

$$\frac{1}{3} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

SOUTH

Double edges:

$$\frac{1}{2} \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

NORTH-EAST

$$\frac{1}{2} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

NORTH-WEST

$$\frac{1}{2} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

SOUTH-EAST

$$\frac{1}{2} \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

SOUTH-WEST

$$\frac{1}{2} \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

NORTH-SOUTH

$$\frac{1}{2} \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

EAST-WEST

The convolution masks for the exemplary filter are applied to the target pixel and its neighbourhood of pixels dependent upon the number and particular combination of edge sites detected as "ON".

2.2.5. Flow Diagram of Smoothing Process

Fig. 7 is a flow diagram providing an overview of the process of smoothing edges contained in image data in accordance with the first embodiment. The process of Fig. 7 is preferably used to implement step 620 of Fig. 6A for applying smoothing to a target pixel dependent upon edge data.

The process of Fig. 7 is illustrated for the processing of a single target pixel. However, this process can be readily repeated using stored edge data as a post edge

detecting process. Alternatively, the process of Fig. 6A could be modified by placing step 620 between steps 616 and 618 so that the process can be implemented on a per target pixel basis as the edge detection process is carried out for each target pixel of image data.

5 Processing commences in step 700. In step 702, the currently processed target pixel is checked to determine if all edge sites of the target pixel are off. If decision block 702 returns true (yes), processing continues at decision block 704. In decision block 704, a check is made to determine if the absolute value of the horizontal and vertical gradients of the target pixels are greater than the threshold gradient. If decision block 704 returns true (yes), processing continues at step 708. In step 708, full smoothing is applied to the target pixel in accordance with rule 1 of Table 2. Otherwise, if decision block 704 returns false (no), processing continues at step 706. In step 706, no smoothing is applied to the target pixel in accordance with rule 1. After step 706 or 708, processing continues at step 710. In step 710, the target pixel processed in accordance with the smoothing method of the first embodiment is returned.

10 If decision block 702 returns false (no), processing continues at decision block 712. In decision block 712, a check is made to determine if there is more than one edge site of the target pixel that is turned on. If decision block 712 returns false (no), processing continues at step 714. In step 714, the filter values of the convolution mask are modified for the case of a single edge. In particular, the filter values are modified in accordance with rule 2 of Table 4 dependent upon whether the single edge site is the East, North, West or South edge site. Processing then continues at step 718. Otherwise, if decision block 712 returns true (yes), processing continues at step 716. In step 716, the filter values for the convolution mask are modified for multiple detected edges. The filter coefficients are modified in accordance with rule 3 of Table 4, which lists the six possible combinations for a double edge. In this connection, each of six possible modifications is to be read so that the two noted edges for each of the six possibilities is read conjunctively (e.g., north-east is the north and east edge site).

20 In step 718, the modified filter is applied to the target pixel and the surrounding neighbourhood pixels to generate the filtered target pixel. Table 5 illustrates modified filter values for the exemplary filter of Table 3 produced by respective steps 714 and step 716 of Fig. 7 dependent upon the "ON" edge site(s). In step 710, the process target pixel is returned.

2.3. Example of Edge Detection and Smoothing Processes

Fig. 8A is a block diagram of an array 800 of input pixels containing a jagged or unsmoothed diagonal line 802 having a slope of 45 degrees. In this example, the pixels of the input image have either black (0) or white (1) intensity levels. However, intermediate grey-scale values are possible. The processing of all of the pixels is described by way of example with reference to a number of pixels 810-816 and 824 in terms of the edge detection process of Figs. 6A-6C and the smoothing process of Fig. 7. The edge detection results 840 and the smoothed image data 870 are shown in Figs. 8B and 8C, respectively.

To detect edges in the image data array 800, the process of Fig. 6A is applied to the input array 800. In this case, because the input image is greyscale, only a single "colour" component is processed for each target pixel and its neighbours.

Fig. 8B illustrates the results 840 of the edge detection process of Figs. 6A-6C applied to the input data array 800. For example, the edge detection result 850 corresponding to input pixel 810 of Fig. 8A indicates there are zero edges that are "ON". As shown in Fig. 8A, the target pixel 810 does not have any possible "ON" edges since the target pixel 810 and each of its neighbouring pixels surrounding a respective edge site are white. The pixels have the same intensity as the target pixel, and therefore the edge sites are all "OFF". In contrast, the next pixel 812 when processed as the target pixel has a South edge that is "ON". This is due to the intensity difference between the target pixel 812 (white) and the bottom pixel 816 (black) exceeding the predetermined intensity threshold (Tf). Also, the vertical gradients ($G_y = -3/4$ and $G_y = -3/4$) of the target pixel 812 and the bottom pixel 816 are less than the predetermined threshold ($T_g = -1/3$). Further, the vertical gradients ($G_y = 0$ and $G_y = -1/4$) of the respective diagonal pixels 818 and 820 are greater than the predetermined threshold ($T_g = -1/3$). Thus, the edge detection result 852 for the input pixel 812 indicates that the South (S) edge is "ON". This type of processing is continued for the entire array 800 of pixels of Fig. 8A. Similarly, input pixel 814 has its East edge "ON" as indicated by "E" in result 854 in Fig. 8B.

The edge detection processing applied to pixel 816 determines that this pixel has zero edges that are "ON" as indicated by edge detection result 856 in Fig. 8B. Pixel 816 (black) has the same intensity as its right and bottom neighbouring pixels 824, 828, and therefore these edge sites are "OFF". However, the intensities of the top and left pixels 812, 814 (both white) are sufficiently different to that of the target pixel 816 so that the relevant intensity differences exceed the predetermined intensity threshold (Tf). The top pixel 812 and the target pixel 816 have adequate vertical

gradients ($G_y = -3/4$ and $G_y = -3/4$) lower than the relevant vertical gradient threshold ($T_g = -1/3$). However, the relevant diagonal pixels 826, 830 are such that the vertical gradient ($G_y = -1/2$) of the diagonal pixel 826 is lower than the vertical gradient threshold (the gradient of diagonal 830 does satisfy the gradient requirement with $G_y = 3/4$). Thus, the North edge site is "OFF". The intensity of the left pixel 814 (white) also satisfies the intensity difference requirement for a West edge site and the horizontal gradients ($G_x = -3/4$) of pixels 814 and 816 are lower than the predetermined threshold ($G_x = -1/3$). However, the relevant diagonal pixels 822, 830 are such that the horizontal gradient ($G_x = -1/2$) of the diagonal pixel 822 is lower than the horizontal gradient threshold ($G_x = -1/3$); the gradient of diagonal 830 does satisfy the gradient requirement with $G_x = 3/4$. Accordingly, the West edge site is also "OFF". Thus, all edges of target pixel 816 are "OFF".

With regards to pixel 824 as the target pixel, the left pixel 816 and the bottom pixel 830 have the same intensity (all black), and therefore the West and South edge sites are "OFF". In contrast, the top and right pixels 822, 832 are both white, and therefore the intensity difference exceeds the predetermined intensity threshold (T_i) in both cases. Further, the vertical gradients ($G_y = -3/4$ and $G_y = -3/4$) of the top pixel 822 and the target pixel 824 are lower than the predetermined vertical gradient threshold ($T_g = -1/3$), and the horizontal gradients ($G_x = 1/2$ and $G_x = 1/2$) of the right pixel 832 and target pixel 824 are greater than the gradient threshold ($T_g = 1/3$). Also, the vertical gradients ($G_y = 1/2$ and $G_y = 1/4$) of diagonal pixels 828, 836 in relation to the North edge site are greater than the vertical gradient threshold ($T_g = -1/3$), and the horizontal gradients ($G_x = -1/4$ and $G_x = -1/2$) of diagonal pixels 812 and 828 in relation to the East edge site are lower than the horizontal gradient threshold ($G_x = 1/3$). Thus, both the North and East edge sites of the target pixel 824 are "ON" as indicated by "NE" for edge detection result 858. Similar edge detection processing in accordance with Figs. 6A-6C is carried out on the remaining pixels to yield the edge detection results 840.

Smoothing of the jagged line 802 of Fig. 8A is accomplished by turning some pixels to an intermediate colour or greyscale dependent upon the detected edge results 840 using the exemplary, specific filter of Table 3. The smoothing process of Fig. 7 is applied to the input data 800, where the applied smoothing is dependent upon the edge detection results 840 in accordance with Table 5. For example, input data pixel 810 has an edge detection result of "0", and the absolute values of the vertical and horizontal gradients ($G_y = G_x = -1/4$) of pixel 810 are each less than the respective gradient threshold ($T_g = 1/3$). Thus, no smoothing is applied to the target pixel 810 (per steps 702, 704 and 706 of Fig. 7), and the output value 880 of the smoothing processing is that of the input value (i.e., white) of pixel 810. The edge

detection result 852 for input pixel 812 indicates that the South edge only is "ON", and therefore the single-edge, South convolution mask of Table 5 is applied to the pixel 812 (per steps 702, 712, 714 and 718 of Fig. 7). The top, left and right pixels of target pixel 812 are each white (1), and therefore the output result 882 ($= 1/3 * 3$) is white. Likewise, smoothing processing for pixel 814 having the edge detection result 854 of "E" produces a white pixel (again, per steps 702, 712, 714 and 718 of Fig. 7) as output pixel 884.

Input data pixel 816 has an edge detection result 856 of "0", and the absolute values of the vertical and horizontal gradients ($G_y = G_x = -3/4$) of pixel 816 are each lower than the respective gradient threshold ($T_g = -1/3$). Thus, full smoothing is applied to the target pixel 816 (per steps 702, 704, 708 of Fig. 7) using the convolution mask of Table 3. This produces an output greyscale value 886 of $1/2$ (indicated by hatching in Fig. 8C). Input data pixel 824 as target pixel has an edge detection result 858 of NE. Thus, the double-edged, North-East convolution mask of Table 5 is applied to the neighbourhood of pixels around the target pixel 824 (per steps 702, 712, 716 and 718 of Fig. 7). Only pixels 816 and 830 (both of which are black) contribute to the output value 888 ($= 1/2 * 0$). Thus, the output value 888 of this pixel 824 is black.

This smoothing process is carried out on all pixels 800 of Fig. 8A to produce the output pixels 870 of Fig. 8C having the smoothened diagonal line 892. As shown in Fig. 8C, the intensity of the hatched pixels is the average of the background and foreground colours. Thus, if the input image is black and white, the output image contains most likely greyscale values. The visual appearance of diagonal line 892 is significantly improved from the input diagonal line 802 in accordance with the edge detection and smoothing processes of the first embodiment.

2.4. Post-Processing to Preserve High-Contrast Textures

The smoothing process of the first embodiment modifies high-contrast textures, like checkerboards or textures generated by a dithering process. The smoothened image can be post-processed to avoid this effect. Fig. 9A is a flow diagram illustrating generalised steps of post-processing in accordance with the first embodiment. Processing commences in step 900. The post-processing is based on comparing the smoothened image to the original image in step 902 and undoing some of the modifications caused by the smoothing method in step 904. Processing terminates in step 906. The smoothened image is scanned in raster order, and, for each target pixel, a limited neighbourhood of pixels is considered. The target pixels are the pixels having a colour that has been changed by the smoothing process. As

FLCD20 385682 CFP.mkn

le:\claral\cdnew\lfd20\385682.doc

described hereinafter with reference to Figs. 9B to 9D, different kinds of post-processing can be considered.

Fig. 9B is a flow diagram of a first post-processing method that limits the number of modified pixels in a restricted area. This process is applied to each target pixel (i.e., modified pixel value) in the smoothened image. For each target pixel, processing commences in step 920. In step 922, the target pixel and its neighbouring pixels are selected. In step 924, the number m of modified pixels in the predetermined neighbourhood centred on the target pixel of smoothened data is determined. In decision 926, a check is made to determine if more than a predetermined number n of pixel values have been modified by the smoothing process. If decision block 926 returns false (no), processing continues at step 930 and processing terminates. Otherwise, if decision block 926 returns true (yes), processing continues at step 928. In step 928, the target pixel is restored to its original input value. Processing then continues at step 930. Preferably, the number n of predetermined pixels is four. Thus, for each target pixel, if the number of modified pixels in the 3×3 neighbourhood is greater than 4, the original colour of the pixel is restored. Other values for the predetermined number n may be practiced without departing from the scope and spirit of the invention.

Fig. 9C is a flow diagram illustrating a further post-processing procedure. Processing commences in step 940 for each target pixel. In step 942, the target pixel and its neighbouring pixels are selected. In decision block 944, a check is made to determine if the modified pixels conform to a predetermined pattern. If decision block 944 returns true (yes), processing continues at step 948 and processing terminates. Otherwise, if decision block 944 returns false (no), processing continues at step 946. In step 946, the target pixel is restored to its original input value. Thus, the pattern of modified pixels in the smoothened can be considered as well. A change in a pixel value is allowed only if the modified pixels in the neighbourhood are ordered in a certain way. Fig. 10A shows examples of patterns 1010, 1020 and 1030 that may be allowed. The black pixels represent the locations of pixels in a 3×3 neighbourhood having values that are modified. Optionally, the process of Fig. 9C may be implemented in a modified form that is a combination of Figs. 9B and 9C.

A further post-processing method of the first embodiment aims at avoiding important changes in colour balance. Fig. 9D is a flow diagram illustrating this process. Processing for each target pixel commences in step 960. In step 962, the target pixel and its neighbouring pixels are selected. In step 964, the number of neighbouring modified pixels having increased values is determined. In step 966, the number of neighbouring modified pixels having decreased values is determined. In decision block 968, a check is made to determine if the number of increased pixel

values is substantially equal to or equal to the number of decreased pixel values. If decision block 968 returns true (yes), processing continues at step 972 and terminates. Otherwise, if decision block 968 returns false (no), processing continues at step 970. In step 970, the original value of the target pixel is restored. Processing then continues at step 972. Thus, a pixel modification is accepted only if in its neighbourhood (e.g. 3x3 pixels) the number of pixels having increased values is almost equal to the number of pixels having decreased values (e.g., they differ by less than 2). Fig. 10B shows examples of textures that are preserved if the post-processing of Fig. 9D is applied.

3. Thickening Process of Second Embodiment

If the smoothened data is later optionally halftoned for display, the halftoning may cause a problem with the smoothened data by turning intermediate pixels to the background colour. Consequently, text and graphics can look thinner, and the smoothing effect may be less perceptible than before halftoning.

The second embodiment is a modification of the smoothing method according to the first embodiment and it produces a different smoothing effect. This smoothing method has been modified to deal with thin features in the input data. The process of the second embodiment aims at improving the appearance and readability of thin features by thickening them. Advantageously, the smoothing effect achieved by the process of the second embodiment is insensitive to posterior halftoning. However, the process relies on the assumption that the background intensity is greater than the foreground intensity (e.g. black text on a white background). The process can be readily modified to deal with the opposite case (i.e., when the background intensity is lower than the foreground intensity). Thus, the thickening method can be applied to arbitrary data provided minimal colour information is available. This second embodiment also describes a method to deduce whether the foreground is darker or brighter than the background.

3.1. Edge Detection Process

3.1.1. Gradient Computation and Thresholding

In the second embodiment, gradient computation and thresholding are performed as described hereinbefore in section 2.1.1 with reference to the first embodiment.

3.1.2. Edge Site Determination Process

The edge detection process of the second embodiment is substantially the same as that shown in Figs. 6A and 6B. However, modifications are made to the flow diagram of Fig. 6C since different rules are applied to turn on edge sites. The edge sites are turned on if they belong to either a horizontal or vertical line regardless of the length of the line.

A vertical edge site is turned on if the following conditions are met:

- 10 A) the intensities of the two pixels surrounding the edge site (i.e., the target pixel and either the left or right pixel) are significantly different; and
- B) the horizontal gradient values of the two surrounding pixels are an absolute local maxima.

15 Again, referring to Fig. 4, a group of pixels 300 centred on a pixel 400 is shown including the surrounding pixels 400 and 402 that are considered for an eastern edge site 404.

Similarly, the horizontal edge site is turned on if the following conditions are met:

- 20 A) the intensities of the two pixels surrounding the edge site (i.e., the target pixel and either the top or bottom pixel) are significantly different; and
- B) the vertical gradient values of the two surrounding pixels are an absolute local maxima.

25 A more precise statement of the rules for detecting if edge sites are on is given in Table 6 (using the same naming conventions as the first embodiment).

TABLE 6

- 30 Rule 1: The EAST edge is "ON" if:
 - (i) $(\text{abs}((F(i,j) - F(i,j+1))) > T_f)$, AND
 - (ii) $((G_x(i,j) > T_g \text{ and } G_x(i,j+1) > T_g)$
OR
 $(G_x(i,j) < -T_g \text{ and } G_x(i,j+1) < -T_g))$.
- 35 Rule 2: The NORTH edge is "ON" if:
 - (i) $(\text{abs}((F(i,j) - F(i-1,j))) > T_f)$
 - (ii) $((G_y(i,j) > T_g \text{ and } G_y(i-1,j) > T_g)$

FLCD20 385682 CFPunkn

le:\csar\fednew\fld20\385682.doc

OR

$(Gy(i,j) < -Tg \text{ and } Gy(i-1,j) < -Tg))$.

Rule 3: The WEST edge is "ON" if:

- 5 (i) $(\text{abs}((F(i,j) - F(i,j-1))) > Tf)$
 (ii) $((Gx(i,j) > Tg \text{ and } Gx(i,j-1) > Tg)$
 OR
 $(Gx(i,j) < -Tg \text{ and } Gx(i,j-1) < -Tg))$.

Rule 4: The SOUTH edge is "ON" if:

- 10 (i) $(\text{abs}((F(i,j) - F(i,j+1))) > Tf)$
 (ii) $((Gy(i,j) > Tg \text{ and } Gy(i,j+1) > Tg)$
 OR
 $(Gy(i,j) < -Tg \text{ and } Gy(i,j+1) < -Tg))$.

15 The process of determining or detecting special cases of double-edge pixels (referred to as "thickening pixels") is described hereinafter with reference to the smoothing process of the second embodiment.

3.1.3. Modified Portion of Edge Site Determination Process

20 The edge detection process of the second embodiment includes the processes of Figs. 6A and 6B, described hereinbefore, and the process of Fig. 11 to implement the rules of Table 6. Fig. 11 illustrates the generalised steps implemented in each of steps 634 to 640 of Fig. 6B.

25 Processing for each target pixel commences in step 1100. In decision block 1102, a check is made to determine if a difference between the intensity values of the target pixel and the relevant other surrounding pixel (i.e., right, top, left or bottom pixel for the East, North, West or South edge site, respectively) is greater than the predetermined intensity threshold Tf. Decision block 1102 implements sub-rule (i) of each of the rules of Table 6. If decision block 1102 returns false (no) indicating that the absolute value of the intensity different is not greater than the threshold intensity,
30 processing continues at step 1108. In step 1108, the respective edge of the target pixel for the selected colour component is set to "OFF". Processing then continues at step 1110 and the edge data is returned to the calling procedure.

35 If decision block 1102 returns true (yes), processing continues at decision block 1104. In decision block 1104, a check is made to determine if the relevant

gradients (i.e., the horizontal or vertical gradient) of the target pixel and the respective other surrounding pixel (i.e., right, top, left or bottom pixel) are either greater than the gradient threshold or less than the negative value of the gradient threshold. This check implements sub-rule (ii) of Table 6. If decision block returns false (no), processing continues at step 1108 as described hereinbefore. Otherwise, if decision block 1104 returns true (yes), processing continues at step 1106. In step 1106, the respective edge (i.e., East, North, West or South) of the target pixel is set as "ON". Processing then continues at step 1110. While the flow diagrams of Figs. 6A, 6B and 11 illustrate specific steps and ordering of steps, it will again be apparent to one skilled in the art that modifications and/or changes to the particular steps and/or ordering of steps can be made to the edge detection process of the second embodiment without departing from the scope and spirit of the invention.

3.2. Smoothing Process

The thickening process of the second embodiment aims at smoothing jagged lines by turning certain background pixels to the foreground colour, instead of introducing intermediate colour or grey levels as is done in the antialiasing version of the first embodiment. Background pixels that are turned to the foreground pixels by this thickening process are referred to hereinafter as thickening pixels.

3.2.1. Thickening Pixel Determination Rules

Thickening pixels are some of the double-edged pixels (i.e., north-east, north-west, south-east, south-west) that belong to thin lines. They can be distinguished from the other double-edge pixels by checking the following conditions set forth in Table 7.

TABLE 7

NORTH-EAST:	$Gx(i-1, j+1) < -Tg$	and	$Gy(i-1, j+1) > Tg$
NORTH-WEST:	$Gx(i-1, j-1) > Tg$	and	$Gy(i-1, j-1) > Tg$
SOUTH-EAST:	$Gx(i+1, j+1) < -Tg$	and	$Gy(i+1, j+1) < -Tg$
SOUTH-WEST:	$Gx(i+1, j-1) > Tg$	and	$Gy(i+1, j-1) < -Tg$

In Table 7, (i,j) are the coordinates of a target pixel that has two adjacent edge sites turned on. The same naming conventions are used as in the process of the first embodiment. The target pixel is a candidate thickening pixel if the conditions set forth in Table 7 are true. Otherwise, the pixel is a standard double edge, which is processed in accordance with rule 3 of Table 4.

3.2.2. Modified Smoothing Process for Thickening Pixels

The smoothing filter (e.g., the filter of Table 3) is modified according to the results of the edge detection in accordance with rules 2 and 3 of Table 4, except for the thickening pixels. Using the same naming conventions, for the thickening pixels, the smoothing filter values which are different from the classical smoothing filter are set forth in Table 8.

TABLE 8

SOUTH-WEST: $V_a = V_b = V_c = V_f = V_i = 0$ and $N = V_d + V_e + V_g + V_h$
 SOUTH-EAST: $V_a = V_b = V_c = V_d = V_g = 0$ and $N = V_e + V_f + V_h + V_i$
 NORTH-WEST: $V_c = V_f = V_g = V_h = V_i = 0$ and $N = V_a + V_b + V_d + V_e$
 NORTH-EAST: $V_a = V_d = V_g = V_h = V_i = 0$ and $N = V_b + V_c + V_e + V_f$

The convolution masks for the thickening pixels are set forth in Table 9.

TABLE 9

The diagram illustrates four 3x3 matrices, each representing a different direction and having a weight of 1/2. The matrices are arranged in a 2x2 grid. The top-left matrix is labeled 'SOUTH-WEST', the top-right 'SOUTH-EAST', the bottom-left 'NORTH-WEST', and the bottom-right 'NORTH-EAST'. Each matrix has a 1/2 weight indicated to its left.

0	0	0
1	0	0
0	1	0

1/2

0	0	0
0	0	1
0	1	0

1/2

SOUTH-WEST

0	1	0
1	0	0
0	0	0

1/2

SOUTH-EAST

0	1	0
0	0	1
0	0	0

1/2

NORTH-WEST

0	0	1
0	0	1
0	0	0

1/2

NORTH-EAST

The effect of the convolution masks for the thickening pixels stated in Table 9 is apparent when compared with the convolution masks for the respective double edges stated in Table 5 for the standard or normal case (i.e., non-thickening cases). The convolution masks for this exemplary filter are applied to the target pixel and its neighbourhood of pixels dependent upon the presence of predetermined "thickening" pixels, in addition to the number and particular combination of edge sites detected as "ON".

3.2.3. Flow Diagram of Smoothing Process

10

Fig. 12 is a flow diagram providing an overview of the process of smoothing edges contained in image data in accordance with the second embodiment. The process of Fig. 12 is preferably used to implement step 620 of Fig. 6A for applying smoothing to a target pixel dependent upon edge data.

15

The process of Fig. 12 is illustrated for the processing of a single target pixel. However, as described hereinbefore with reference to Fig. 7, this process can be readily repeated using stored edge data as a post edge detecting process.

20

Processing commences in step 1200. In decision block 1202, a check is made to determine if all of the edge sites of the target pixel are "OFF". If decision block 1202 returns true (yes), processing continues at step 1216 and the processed target pixel is returned. In this case, the value of the processed target pixel is equal to its input value. Otherwise, if decision block 1202 returns false (no), processing continues at decision block 1204. In decision block 1204, a check is made to determine if more than one edge site is "ON". If decision block 1204 returns false (no), processing continues at step 1208. In step 1208, the convolution mask is modified for a single edge in accordance with rule 2 of Table 4. Processing then continues at step 1214. In step 1214, the modified filter is applied to the target pixel. In step 1216, the processed target pixel produced by step 1214 is returned to the calling process.

30

Otherwise, if decision block 1204 returns true (yes) indicating the target pixel has at least two "ON" edge sites, processing continues at decision block 1206. In decision block 1206, a check is made to determine if the target pixel is a thickening pixel in accordance with the rules of Table 7 for North-East, North-West, South-East, and South-West cases. If decision block 1206 returns false (no), processing continues at step 1210. In step 1210, the filter is modified for a "standard" (i.e., normal case) double edge site in accordance with the rule 3 of Table 4. In the case that the exemplary filter of Table 3 is used, the filter is modified so as to produce or use the respective double-edge convolution mask of Table 5. Processing then continues at

35

FLCD20 385682 CFPunkn

le:\cira\afcdnew\fld20\385682.doc

step 1214, as described hereinbefore. Otherwise, if decision block 1208 returns true (yes) indicating a thickening pixel, processing continues at step 1212. In step 1212, the filter is modified in accordance with the rules of Table 8. For the case of the exemplary filter of Table 3, the convolution masks of Table 9 are produced or used. Processing then continues at step 1214, as described hereinbefore. Thus, the process of Fig. 12 applies modified convolution masks to the target pixel and its neighbourhood of pixels dependent upon the presence of predetermined "thickening" pixels, in addition to the number and particular combination of edge sites detected as "ON".

3.2.4. Example of Edge Detection and Smoothing Process

Fig. 13A is a block diagram of an array 1300 of input pixels containing a jagged or unsmoothed diagonal line 1302 having a slope of 45 degrees. This pixel array 1300 is identical to the array 800 of Fig. 8A. The processing of a number of pixels 1304-1310 is described in terms of the edge detection process of Figs. 6A, 6B and 11 and the smoothing process of Fig. 12. The processing is applied to the entire array 1300, but is only described with reference to certain pixels for the purpose of brevity. The edge detection results 1340 and the smoothed image data 1370 are shown in Figs. 13B and 13C, respectively. The results of the edge detection process are described with particular reference to results 1354-1360 for pixels 1304-1310, respectively, to point out the effect of thickening pixels in the smoothing process of the second embodiment. The edge detection rules of Table 6 for the second embodiment produces substantially the same results as those obtained in Fig. 8B for the first embodiment, except the diagonal row of pixels above and below the central diagonal result line (e.g. result 856 above the central diagonal line containing result 858) having "ON" edge values of "0" in Fig. 8B now have double edge combinations (e.g. 1354 and 1356 above the central diagonal, and 1358 and 1360 below) in Fig. 13B.

Edge detection results 1354 and 1358 (i.e., NW and SE) above and below the central diagonal line are determined to be standard or normal double-edge sites in accordance with the process of Fig. 11 and the rules of Table 6 and since they do not satisfy the thickening rules of Table 7. In contrast, the double-edge site results 1356 and 1360 (i.e., NW and SE) satisfy the relevant thickening rules of Table 7 and therefore are thickening pixels, as indicated by underlining in Fig. 13B. In accordance with steps 1202, 1204, 1206, 1210 and 1214, the standard double edge results 1354 and 1358 causes pixels 1304 and 1308 to be smoothed using the convolution mask for NW and SE edges in accordance with Table 5. In contrast, the thickening results 1356 and 1360 cause pixels 1306 and 1310 to be smoothed in

FLCD20 385682 CFPunkin

le:\ccralfcdnew\fld20\385682.doc

accordance with the respective modified convolution masks of Table 9. The differing effects between standard and thickening double-edge sites is readily determined by comparing the convolution masks for NW, SW, SE, and NE edge sites of Table 5 with the masks for NW, SW, SE, and NE sites in Table 9. The effect of smoothing for thickening pixels is to convert the background colour (white) of input pixels 1306 and 1310 to the foreground colour (black) for output pixels 1386 and 1390 of the output array 1370. In contrast, the standard double-edge input pixels 1304 and 1308 have the same foreground colour in the respective output pixels 1384 and 1388. Thus, with the specific filter chosen, smoothing of the jagged line in accordance with the process of the second embodiment is accomplished by turning some pixels that belong to the background to the foreground colour, with those pixels corresponding to cases of thickening double edges.

Fig. 14A illustrates English language text 1400 input to the edge detection and smoothing process of the second embodiment. Fig. 14B shows the thickened output text 1450 in accordance with the second embodiment.

Likewise, Figs. 15A to 15D show illustrative results obtained using Japanese language text. Fig. 15A illustrates input text 1500 where the jagged edges of the text are readily apparent. The background colour is primarily white, and the foreground colour of the text is black. However, in contrast, one of the characters 1502 has a black background and the text is white, thereby presenting a special case requiring a variation of the process of the second embodiment. Fig. 15B illustrates the thickened output text 1520 produced in accordance with the process of the second embodiment (without colour detection being performed). This process has resulted in the "reversed" or inverted colour character 1520 becoming more jagged. In Fig. 15C, the region 1542 containing the reversed character is selected from the text region 1540 for separate processing. This is done by zone mapping the input text using colour detection. The process of zone mapping is described hereinafter with reference to Figs. 14 and 15. In the zone map, white areas correspond to a first type of pixel (A) with white background and black foreground. The dark area 1542 is a second type of pixel (B) having opposite colour characteristics. For the case where the background colour is less intense than the foreground colour, the same processes of the second embodiment are applied but with all signs of gradients negated. Fig. 15D illustrates the output text 1560 containing the thickened output 1562 produced by this variation of the second embodiment (involving colour detection). The overall appearance and readability of the text is improved even if individual letters are not always improved.

The smoothing algorithm only modifies thin (ie., two-pixel wide) lines. Wider staircases are left unchanged. With the chosen filter, no intermediate value is introduced and the modifications to the data are unlikely to be cancelled by any

subsequent halftoning procedure. This smoothing method is particularly suitable for small features, because it makes them appear thicker.

3.2.5. Process of Preserving High Contrast Textures

5

If applied as described hereinbefore, the thickening method may adversely affect high contrast textures more than the antialiasing method of the first embodiment. However, it is possible to minimise texture modifications. The preserving process comprises the step of limiting the number of thickening pixels in each 2x2 array of
10 pixels. For each thickening candidate pixel, if other background neighbouring pixels are also candidates for thickening, none of them is turned to the foreground colour. This process can be limited to considering only diagonal neighbours or adjacent and diagonal neighbours in the array 1600 shown in Fig. 16. Pixel 1608 (T) is the pixel in the 2x2 array 1600 likely to be thickened. Pixel 1602 (D) is a diagonal neighbour, and pixel 1604 and 1606 (A) are adjacent neighbours.
15

The smoothing process of the second embodiment shown in Fig. 12 can be modified to implement this variation. In particular, Fig. 20 is a flow diagram illustrating the detailed sub-steps of step 1212 of Fig. 12 to carry out this texture preserving process. The processing of Fig. 12 is carried out as described hereinbefore, except a number of sub-steps are carried out for step 1212. When processing is carried out for step 1212, the texture processing commences in step 2000. In decision block 2002, a check is made to determine if there are other pixels of the target pixel in the predetermined size array that are candidates for thickening. If decision block 2002 return false (no), processing continues at step 2004. In step 2004, the double-edge convolution matrix is modified for a thickening pixel. Processing then continues at step 2008, where it returns to the calling process. Otherwise, if decision block 2002 returns true (yes), processing continues at step 2004 and a standard double-edge mask is applied, but no thickening. Again processing terminates at step 2008.
20
25

Fig. 10B shows examples of dithered textures that are not modified by the
30 thickening method if the number of thickening pixels per 2x2 pixel array is limited.

3.2.6. Process of Colour Detection

Fig. 15A illustrates the effect of simply applying the second embodiment, where a character 1502 is inverted (ie., white text on a black background). However, if the background is darker than the foreground, the thickening process can be applied successfully if all the gradient signs are changed. Thus, the problem can be eliminated by providing the smoothing process with minimal colour information (ie., detecting
35

whether the foreground is darker or brighter than the background). Inverted text is often found on PC screens, for example, when text is selected in an editing application. More generally, input data for the smoothing process can contain any text colour on any background colour, with few or many transitions. For example, such circumstances are often encountered in most pages found on the World Wide Web (WWW).

The process of introducing additional colour information involves zone segmenting an image. The zone-segmentation method of this embodiment distinguishes three types of zones:

- 1) areas where dark features are in a minority;
- 2) areas where bright features are in a minority; and
- 3) areas where dark and bright features are equally represented.

The three types of zones may be present in the same image, with horizontal or vertical transitions.

The zone detection process is based, for example, on local statistics computed on the luminance image. In the exemplary case of using the RGB colour model, the luminance of each pixel is obtained using the following formula: $L = 0.299 * R + 0.587 * G + 0.114 * B$. In the process, each pixel is classified depending on the luminance distribution in its neighbourhood (i.e., depending on the relative importance of dark and bright pixels). The local statistics used are preferably the mean, the minimum and the maximum value of luminance in a predetermined neighbourhood. However, other statistics characterising the luminance histogram may be used (e.g., the median) without departing from the scope of the invention.

Fig. 17A is a flow diagram of the zone segmentation method of the third embodiment, showing the generalised steps of this process. The process comprises three fundamental steps (1702, 1704, 1706). Processing commences in step 1700. In step 1702, a representative window is chosen for each pixel (i.e., transitions are detected). In step 1704, local statistics determined for the window are compared to obtain a zone map. In step 1706, the zone map is smoothed.

The first step 1702 involves determining a representative window in the neighbourhood of each pixel. Choosing a reference window so that the target pixel is the centre of the window may induce errors when the target pixel is located near a zone transition. A homogeneous reference window can be selected for each pixel by comparing different adjacent windows. A similarity criterion must be defined for window comparison. For example, two windows may be considered similar if their local statistics are similar (i.e., the absolute difference between their respective minimum, maximum and mean values is smaller than fixed thresholds).

Fig. 17B is a flow diagram illustrating preferred substeps of step 1702 in greater detail. Processing commences in step 1710. In step 1712, two initial windows

are selected for processing, each containing the target pixel centred on a relevant transition. The transition may be either a vertical or horizontal one. In step 1714, statistics of each of the selected windows containing the target pixel are determined. In step 1716, the statistics of the respective pair of windows are compared. In decision block 1718, a check is made to determine if the statistics are similar. If decision block 1718 returns true (yes), processing continues at step 1720. In step 1720, the two windows are merged together to form a new reference window. In step 1722, the merged window is returned.

Otherwise, if decision block 1718 returns false (no), processing continues at decision block 1724. In decision block 1724, a check is made to determine if the size of each of the windows is less than a predetermined size. If decision block 1724 returns true (yes), processing continues at step 1726. In step 1726, no reference windows are determined and processing returns to the calling process. Otherwise, if decision block 1724 returns false (no), processing continues at step 1728. In step 1728, each of the two currently selected windows is split or divided into two further windows. Each pair of these windows is in turn processed. Processing then continues at step 1714.

An example of this processing is illustrated in Figs. 18A to 18E for the case of vertical transitions being detected. Two windows 1810 and 1812 on the left and right of a target pixel 1800 in Figs. 18A and 18B, respectively, are compared (per step 1716 of Fig. 17B). Each of the windows is configured to include the target pixel 1800. If they are similar, both windows 1810 and 1812 are merged to form a reference window 1820, shown in Fig. 18C, for the centre pixel 1800 (per steps 1718, 1720 and 1722).

Otherwise, both windows 1810, 1812 are each split into two equal windows. That is, the window 1810 is split into a left window 1830 and a right window 1832 shown in Fig. 18D, and the window 1812 is split into a left window 1840 and a right window 1842 shown in Fig. 18E (per steps 1724 and 1728). If windows 1830 and 1832 are sufficiently similar, their reunion 1810 of Fig. 18A is considered as the reference window for the target pixel 1800. If windows 1840 and 1842 are similar enough, their reunion 1812 is considered as the reference window for the target pixel 1800. Otherwise, if none of the windows can be chosen, the whole procedure is repeated on smaller windows. For instance, right window 1832 is used instead of window 1810, and window 1840 instead of window 1812. The procedure starts with a chosen window size and stops when a reference window has been determined, or when the current window size becomes inferior to a selected minimal size (per steps 1724 and 1726). Horizontal transitions are detected in a similar fashion, by replacing the left and right windows with top and bottom windows.

In relation to step 1704, the zone segmentation is based on comparing the mean luminance to the minimum and maximum luminance of the reference window and

relies on two thresholds. The variables Min, Max, LT and HT are the minimum and maximum luminance and the lower and higher threshold, respectively. The lower and higher thresholds LT and HT may be chosen using a percentage of the luminance range:

$$LT = \text{Min} + \text{pct} * (\text{Max} - \text{Min}), \text{ and}$$

$$HT = \text{Max} - \text{pct} * (\text{Max} - \text{Min}),$$

where pct = 0.33, for example.

If the mean luminance of a representative window has a value in the range of HT to Max (i.e., case A) as shown in Fig. 19A, dark features are likely to be in the minority. In contrast, if the mean luminance has a value in the range of Min to LT (i.e., case B) as shown in Fig. 19A, bright features are likely to be in the minority. Otherwise, if the mean luminance has a value in the range of LT to HT (i.e., case C) as shown in Fig. 19C, dark and bright features are likely to be equally represented. Thus, at the end of the two first steps 1702 and 1704, four categories of pixels can be distinguished:

- (i) undetermined pixels: pixels for which a reference window could not be chosen, or Min is equal to Max in the reference window;
- (ii) case A: pixels belonging to an area where dark features are to be thickened;
- (iii) case B: pixels belonging to an area where bright features are to be thickened; and
- (iv) case C: pixels belonging to an area where none of the features are to be thickened.

Fig. 17C is a flow diagram illustrating preferred substeps of step 1704 in greater detail. Processing commences in step 1750. In step 1752, the mean, Min and Max luminance and lower and higher threshold are determined for the reference window. In decision block 1764, a check is made to determine if the Min luminance equals the Max luminance. If decision block 1754 returns true (yes), processing continues at step 1756 and returns to the calling process. This is for the case where pixels are undetermined in terms of colour. Otherwise, if decision block 1754 returns false (no), processing continues at decision block 1758. In decision block 1758, a check is made to determine if the mean luminance of the window is greater than the higher luminance threshold. If decision block 1758 returns true (yes), processing continues at step 1760. In step 1760, the zone map is returned as having dark features in the minority.

Otherwise, if decision block 1758 returns false (no), processing continues at decision block 1762. In decision block 1762, a check is made to determine if the mean luminance is less than the lower luminance threshold. If decision block 1762 returns true (yes), processing continues at step 1764. In step 1764, the zone map is returned as

having bright features in the minority. Otherwise, if decision block 1762 returns false (no), processing continues at step 1766. In step 1766, the zone map is returned as being equally likely to have dark and bright features.

For case A pixels, the smoothing method is applied as described in the second embodiment. For case B pixels, the smoothing method of the second embodiment is applied with all gradient signs being changed beforehand. For case C pixels, preferably, another smoothing method, which does not thicken any colour may be used, or no smoothing at all may be performed.

The next stage of the colour detection method comprises the step 1706 of smoothing the map obtained in step 1704. This last step 1706 aims at eliminating small regions, as well as affecting a label to undetermined pixels based on their neighbourhood (e.g., a 3x3 or 5x5 window).

Fig. 17D is a flow diagram illustrating preferred substeps of step 1706 in greater detail. Processing of a zone map commences in step 1770. In step 1772, a first, commencing target pixel of the zone map is selected. In step 1774, the number NA and NB, or NA, NB and NC of labels (A, B, C) in the predetermined neighbourhood are determined (ignoring undetermined labels in the neighborhood). In step 1776, the largest one of NA, NB or NA, NB, NC is selected. In step 1778, the label of the largest number determined in step 1776 is applied to the target pixel. In decision block 1780, a check is made to determine if all of the pixels in the zone map have been processed. If decision block 1780 returns false (no), processing continues at step 1782. In step 1782, the next target pixel in raster scan order is selected for processing. Processing then continues at step 1774. Otherwise, if decision block 1780 returns true (yes), processing continues at decision block 1784. In decision block 1784, a check is made to determine if a predetermined number of iterations of this zone map smoothing processing have been carried out. If decision block 1784 returns false (no), processing continues at step 1772. Otherwise, if decision block 1784 returns true (yes), processing continues at step 1786 and the smoothed zone map is returned to the calling process.

Thus, the zone map obtained in step 1704 of Fig. 17A is scanned in raster order with each pixel being modified depending on its neighbours' label (i.e., A, B, C, or undetermined). If a label is predominant in the neighbourhood of a centre pixel, this predominant label is applied to the target pixel. The procedure is typically repeated a number of times. The "undetermined" label is not propagated, so that this category is progressively eliminated. The variables NA, NB and NC are the number of type A, B and C pixels, respectively, in the neighbourhood of the currently processed pixel. Two options for determining the new label of a target pixel are:

- (1) the maximum value of NA, NB, and NC determines the new label of the centre pixel; and

(2) the maximum value of NA, NB determines the new label of the centre pixel.

Choosing the second option instead of the first option reduces the number of pixels in category C, so that most features are thickened. A further option is to do first a few iterations with option (1) propagating labels A, B and C, and then a few iterations with option (2) propagating only labels A and B.

Fig. 15D shows the results obtained on the Japanese character image 1500 when colour detection is performed before smoothing, as described hereinbefore. The parameters for the colour detection method employed in this example are:

window selection:

height = 5,
maximum width = 25,
minimum width = 5; and

map smoothing:

5x3 neighbourhood,
20 type 1 iterations
followed by 5 type 2 iterations.

4. Combined Antialiasing and Thickening Process of Third Embodiment

The third embodiment of the invention takes advantage of both antialiasing and thickening algorithms by selectively applying either of them for each pixel. This combined method aims at thickening thin foreground features and antialiasing other jagged edges.

The colour detection method described hereinbefore performs zone segmentation which classifies each pixel into one of the three categories and a different smoothing method is applied to each of these categories. The three categories are:

- A) areas where dark features are in a minority; the thickening method is applied to thicken dark oblique lines;
- B) areas where bright features are in a minority; the thickening method is applied after all the gradient signs are changed to thicken bright oblique lines;
- C) areas where dark and bright features are equally represented; the antialiasing method is applied to smooth any oblique edge without thickening any colour.

This combined process of the third embodiment allows thin foreground features to be thickened, except when they are too dense, which makes them more

readable. This is the case especially after halftoning. Using the thickening method of the second embodiment when no colour is in a minority can affect dense text by connecting characters which should not be connected. Thus, it is preferable to use the antialiasing method for the third category.

5 Furthermore, to take advantage of the full potential of the combination of the antialiasing and thickening methods of the first and second embodiments, the output of both methods can be compared. The thickening method modifies only thin features (e.g., one XGA pixel-wide lines), and isolated edges are left unchanged. Therefore, wide characters located in type A or B area or zone is not smoothed. The antialiasing
10 algorithm, on the other hand, smooths any jagged edge. Hence, there is a need to detect edges that are modified by the antialiasing process but not by the thickening process in zones A and B.

If an edge is antialiased, the difference between the antialiased image 2210 shown in Fig. 22B and original (or thickening processed) image 2200 shown in Fig. 22A illustrates alternate signs on a diagonal line 2222 of the difference map 2220
15 shown in Fig. 22C. Thus, a pixel belonging to zone A or B that should be antialiased is detected by comparing the sign of the difference between the antialiased and thickened images for the current pixel and its diagonal neighbours. If for one diagonal neighbour, the sign of the difference differs, the centre pixel is antialiased. Because of this
20 comparison, all of the jagged edges are smoothed.

4.1 Flow Diagram of Process of Third Embodiment

Fig. 21 is a flow diagram illustrating the combined antialiasing and thickening method of the third embodiment. Processing of the input data commences
25 in step 2100. In step 2102, the colour detection process is applied to the image to detect zones A, B, C in the input data. This is done in accordance with the process described hereinbefore with reference to Fig. 17D. In step 2104, a first one of the detected zones is selected for processing. In decision block 2106, a check is made to
30 determine if dark or bright features are in the minority in the selected zone. If decision block 2106 returns false (no) indicating a zone of type C, processing continues at step 2108. In step 2108, the antialiasing edge detection and smoothing processing of the first embodiment is applied to the zone. Processing then continues at decision block 2118.

35 Otherwise, if decision block 2106 returns true (yes) indicating a zone of type A or B, processing continues at step 2110. In step 2110, the thickening edge detection and smoothing process is applied to the selected zone. In step 2112, the antialiasing edge detection and smoothing processing of the first embodiment is also

applied to the zone. In step 2114, the results of the antialiasing and thickening processing are compared dependent on the sign of differences between the antialiased and thickened results. In step 2116, detected antialiasing results for differences between the thickening and antialiasing results are incorporated in the thickening results. Processing then continues at decision block 2118.

5 In decision block 2118, a check is made to determine if there are more zones to be processed. If decision block 2118 returns true (yes), processing continues at step 2122. In step 2122, the next zone is selected, and processing continues at decision block 2106. Otherwise, if decision block 2118 returns false (no), processing
10 terminates at step 2120.

The foregoing describes only a small number of embodiments of the present invention. Modifications and/or changes, obvious to one skilled in the art, can be made thereto without departing from the spirit and scope of the present invention.

FIG. 10

The claims defining the invention are as follows:

1. A method of smoothing jagged edges in graphical data, said method comprising the steps of:

detecting one or more edges of a selected pixel in said graphical data dependent upon intensities of said selected pixel and another pixel surrounding a respective site of said one or more edges and predetermined gradients of at least said selected pixel and said respective site surrounding pixel; and

applying an adapted convolution mask to said selected pixel and a predetermined neighbourhood of pixels containing said selected pixel, wherein coefficient values of said convolution mask are dependent upon said one or more detected edges.

2. The method according to claim 1, wherein the detecting step is dependent upon a difference between the intensities of said selected pixel and said site surrounding pixel satisfying an intensity threshold.

3. The method according to claim 2, wherein the detecting step is dependent upon an absolute value of the difference exceeding said intensity threshold.

4. The method according to claim 1, wherein the detecting step is dependent upon the predetermined gradients of said selected pixel and said site surrounding pixel satisfying a gradient threshold.

5. The method according to claim 4, wherein the predetermined gradients are either vertical or horizontal dependent upon said edge site being either horizontal or vertical, respectively.

6. The method according to claim 4, wherein the predetermined gradients are greater than said gradient threshold.

7. The method according to claim 4, wherein the predetermined gradients are less than said gradient threshold.

8. The method according to claim 4, wherein the detecting step is further dependent upon predetermined gradients of diagonal pixels in said neighbourhood satisfying a gradient threshold.

9. The method according to claim 8, wherein:
the predetermined gradients of said selected pixel and said site surrounding pixel are greater than said gradient threshold; and
the predetermined gradients of said diagonal pixels are less than said gradient threshold.

10. The method according to claim 8, wherein:
the predetermined gradients of said selected pixel and said site surrounding pixel are less than said gradient threshold; and
the predetermined gradients of said diagonal pixels are greater than said gradient threshold.

11. The method according to claim 1, wherein if a single edge is detected, one or more coefficient values of a portion of said convolution mask corresponding to an edge containing said site surrounding pixel is set to a predetermined value.

12. The method according to claim 11, wherein said predetermined value is zero.

13. The method according to claim 1, wherein if a double edge is detected, one or more coefficient values of each portion of said convolution mask corresponding to a respective edge containing one of the two site surrounding pixels are set to a predetermined value.

14. The method according to claim 13, wherein said predetermined value is zero.

15. The method according to claim 6 or 7, further comprising the step of, for a detected double edge, determining whether said selected pixel is a thickening pixel dependent upon a horizontal gradient and a vertical gradient of a pixel at the junction of the two edges satisfy respective gradient thresholds.

16. The method according to claim 15, wherein one or more coefficient values of a portion of said convolution mask corresponding to each edge opposite to the detected double edges is set to a predetermined value.

17. The method according to claim 1, wherein said detecting step is carried out for each colour component of said selected pixel, and said one or more detected edges are dependent upon the largest intensity and gradient values of the colour components.

18. A method of smoothing jagged edges in graphical data, said method comprising the steps of:

25 determining one or more edges if the intensities of a target pixel and a neighbouring pixel, both abutting a respective edge site, are substantially different, predetermined gradients dependent upon said respective edge site are an absolute local maxima, and predetermined gradients of diagonal pixels are not an absolute local maxima of the same sign as that of the neighbouring pixel; and

30 modifying said target pixel using a filter having coefficient values dependent upon said one or more determined edges.

19. The method according to claim 18, further comprising the step of setting to zero one or more coefficient values of a portion of said filter corresponding to each detected edge.

35 20. The method according to claim 18, further comprising the steps of: comparing said modified value of said target pixel with the original value of said target pixel.

restoring said target pixel to its original value dependent upon number of modified values of said graphical data.

21. The method according to claim 18, further comprising the steps of:

5 determining the number of modified pixels in a predetermined neighbourhood of pixels containing said target pixel; and

if said number exceeds a threshold number of modified pixels, restoring one or more modified pixels in said neighbourhood to the original value of the respective pixel.

22. The method according to claim 18, further comprising the steps of:

10 determining if modified pixels in a predetermined neighbourhood of pixels containing said target pixel conform to a predetermined pattern of pixels; and

restoring any pixels determined not to conform to the predetermined pattern to the original value of the respective pixel.

23. The method according to claim 18, further comprising the steps of:

15 determining the number of modified pixels in a predetermined neighbourhood of pixels containing said target pixel having increased values;

determining the number of modified pixels in a predetermined neighbourhood of pixels containing said target pixel having decreased values; and

20 restoring the original value of said target pixel if the number of increased-value modified pixels is not substantially the same as the number of decreased-value modified pixels.

24. A method of smoothing jagged edges in graphical data, said method comprising the steps of:

25 determining one or more edges if the intensities of a target pixel and a neighbouring pixel, both abutting a respective edge site, are substantially different, and predetermined gradients dependent upon said respective edge site are an absolute local maxima; and

modifying said target pixel using a filter having coefficient values dependent upon said one or more determined edges.

30 25. The method according to claim 24, further comprising the step of setting to zero one or more coefficient values of a portion of said filter corresponding to each detected edge.

26. The method according to claim 24, further comprising the steps of:

determining if a target pixel is a thickening pixel; and

35 setting to zero one or more coefficient values of a portion of said filter corresponding to an edge opposite to each detected edge.

27. The method according to claim 26, wherein said target pixel is determined to be a thickening pixel if said target pixel has two edges and if the

horizontal and vertical gradients of a pixel at the junction of said two edges and diagonally abutting said target pixel are absolute local minima.

28. The method according to claim 27, further comprising the steps of:
determining if pixels in a predetermined size neighbourhood of said target
5 pixel are also candidates for thickening; and
if said neighbouring pixels are also thickening candidates, restoring said target pixel to its original value.

29. The method according to claim 23, further comprising the steps of:
determining a zone containing said neighbourhood of pixels; and
10 if said zone is an inverted zone having a dark background colour and a bright foreground colour, negating said predetermined gradients of said target and neighbouring pixels.

30. The method according to claim 29, wherein said step of determining a zone comprises the further steps of:
15 choosing a representative window for said target pixel;
comparing local statistics regarding luminance of pixels in the representative window to obtain zone; and
smoothing the zone.

31. A method of smoothing jagged edges in graphical data, said method
20 comprising the steps of:
a) detecting a plurality of zones of said graphical data dependent upon the values of a plurality of pixels contained in each zone, said zones each containing a minority of either bright or dark pixels or containing an equal number of bright and dark pixels; and

25 b) for each zone, filtering pixels of said zone, said zone filtering comprising the steps of:

if said bright and dark pixels are substantially equally present, applying antialiasing to said zone;

30 if said dark pixels are in a minority in said zone, applying thickening to said zone; and

if said bright pixels are in a minority in said zone, applying thickening to said zone with relevant gradients negated.

32. The method according to claim 31, wherein said antialiasing step
35 further comprises, for each pixel of said zone processed as a target pixel, the steps of:
determining one or more edges if the intensities of a target pixel and a neighbouring pixel, both abutting a respective edge site, are substantially different, predetermined gradients dependent upon said respective edge site are an absolute local

FLCD20 385682 CFPunk

lo:\clarat\ednew\flcd20\385682.doc

maxima, and predetermined gradients of diagonal pixels are not an absolute local maxima of the same sign as that of the neighbouring pixel; and

modifying said target pixel using a filter having coefficient values dependent upon said one or more determined edges.

5 33. The method according to claim 31, wherein said thickening step further comprises, for each pixel of said zone processed as a target pixel, the steps of:

determining one or more edges if the intensities of a target pixel and a neighbouring pixel, both abutting a respective edge site, are substantially different, and predetermined gradients dependent upon said respective edge site are an absolute
10 local maxima; and

modifying said target pixel using a filter having coefficient values dependent upon said one or more determined edges.

34. The method according to claim 31, wherein, if said dark or bright pixels are in the minority in said zone, further performing the steps of:

15 applying antialiasing to said zone;

comparing said antialiased zone with said thickened zone; and

replacing values of one or more pixels in said thickened zone with antialiased values of a corresponding pixel in said antialiased zone dependent upon results of said comparison.

20 35. The method according to claim 34, wherein:

said comparing step further comprises generating a difference map by subtracting values of pixels in said thickened zone from corresponding values in said antialiased zone, or vice versa; and

25 determining pixels of said thickened zone corresponding with non-zero values of said difference map and replacing said pixels of said thickened zone with corresponding values of said antialiased zone.

36. The method according to claim 31, further comprising the step of repeating steps (a) and (b) a predetermined number of iterations.

30 37. An apparatus for smoothing jagged edges in graphical data, said apparatus comprising:

means for detecting one or more edges of a selected pixel in said graphical data dependent upon intensities of said selected pixel and another pixel surrounding a respective site of said one or more edges and predetermined gradients of at least said selected pixel and said respective site surrounding pixel; and

35 filtering means for applying an adapted convolution mask to said selected pixel and a predetermined neighbourhood of pixels containing said selected pixel, wherein coefficient values of said convolution mask are dependent upon said one or more detected edges.

38. The apparatus according to claim 37, wherein the detecting operation of the detection means is dependent upon a difference between the intensities of said selected pixel and said site surrounding pixel satisfying an intensity threshold.

39. The apparatus according to claim 38, wherein the detecting operation of the detection means is dependent upon an absolute value of the difference exceeding said intensity threshold.

40. The apparatus according to claim 37, wherein the detecting operation of the detection means is dependent upon the predetermined gradients of said selected pixel and said site surrounding pixel satisfying a gradient threshold.

41. The apparatus according to claim 40, wherein the predetermined gradients are either vertical or horizontal dependent upon said edge site being either horizontal or vertical, respectively.

42. The apparatus according to claim 40, wherein the predetermined gradients are greater than said gradient threshold.

43. The apparatus according to claim 40, wherein the predetermined gradients are less than said gradient threshold.

44. The apparatus according to claim 40, wherein the detecting operation of the detection means is further dependent upon predetermined gradients of diagonal pixels in said neighbourhood satisfying a gradient threshold.

45. The apparatus according to claim 44, wherein:
the predetermined gradients of said selected pixel and said site surrounding pixel are greater than said gradient threshold; and
the predetermined gradients of said diagonal pixels are less than said gradient threshold.

46. The apparatus according to claim 44, wherein:
the predetermined gradients of said selected pixel and said site surrounding pixel are less than said gradient threshold; and
the predetermined gradients of said diagonal pixels are greater than said gradient threshold.

47. The apparatus according to claim 37, wherein if a single edge is detected, one or more coefficient values of a portion of said convolution mask corresponding to an edge containing said site surrounding pixel is set to a predetermined value.

48. The apparatus according to claim 47, wherein said predetermined value is zero.

49. The apparatus according to claim 37, wherein if a double edge is detected, one or more coefficient values of each portion of said convolution mask

corresponding to a respective edge containing one of the two site surrounding pixels are set to a predetermined value.

50. The apparatus according to claim 49, wherein said predetermined value is zero.

51. The apparatus according to claim 42 or 43, further comprising means for determining, for a detected double edge, whether said selected pixel is a thickening pixel dependent upon a horizontal gradient and a vertical gradient of a pixel at the junction of the two edges satisfy respective gradient thresholds.

52. The apparatus according to claim 51, wherein one or more coefficient values of a portion of said convolution mask corresponding to each edge opposite to the detected double edges is set to a predetermined value.

53. The apparatus according to claim 37, wherein the detecting operation of said detection means is carried out for each colour component of said selected pixel, and said one or more detected edges are dependent upon the largest intensity and gradient values of the colour components.

54. An apparatus for smoothing jagged edges in graphical data, said apparatus comprising:

means for determining one or more edges if the intensities of a target pixel and a neighbouring pixel, both abutting a respective edge site, are substantially different, predetermined gradients dependent upon said respective edge site are an absolute local maxima, and predetermined gradients of diagonal pixels are not an absolute local maxima of the same sign as that of the neighbouring pixel; and

filtering means for modifying said target pixel, said filtering means having coefficient values dependent upon said one or more determined edges.

55. The apparatus according to claim 54, further comprising means for setting to zero one or more coefficient values of a portion of said filter corresponding to each detected edge.

56. The apparatus according to claim 54, further comprising:

means for comparing said modified value of said target pixel with the original value of said target pixel.

means for restoring said target pixel to its original value dependent upon number of modified values of said graphical data.

57. The apparatus according to claim 54, further comprising:

means for determining the number of modified pixels in a predetermined neighbourhood of pixels containing said target pixel; and

means for restoring one or more modified pixels in said neighbourhood to the original value of the respective pixel, if said number exceeds a threshold number of modified pixels.

58. The apparatus according to claim 54, further comprising:
means for determining if modified pixels in a predetermined neighbourhood
of pixels containing said target pixel conform to a predetermined pattern of pixels;
and

5 means for restoring any pixels determined not to conform to the
predetermined pattern to the original value of the respective pixel.

59. The apparatus according to claim 54, further comprising:

means for determining the number of modified pixels in a predetermined
neighbourhood of pixels containing said target pixel having increased values;

10 means for determining the number of modified pixels in a predetermined
neighbourhood of pixels containing said target pixel having decreased values; and

means for restoring the original value of said target pixel if the number of
increased-value modified pixels is not substantially the same as the number of
decreased-value modified pixels.

15 60. An apparatus for smoothing jagged edges in graphical data, said
apparatus comprising:

means for determining one or more edges if the intensities of a target pixel
and a neighbouring pixel, both abutting a respective edge site, are substantially
different, and predetermined gradients dependent upon said respective edge site are an
20 absolute local maxima; and

filter means for modifying said target pixel, said filter means having
coefficient values dependent upon said one or more determined edges.

25 61. The apparatus according to claim 60, further comprising means for
setting to zero one or more coefficient values of a portion of said filter corresponding
to each detected edge.

62. The apparatus according to claim 60, further comprising:

means for determining if a target pixel is a thickening pixel; and

means for setting to zero one or more coefficient values of a portion of said
filter corresponding to an edge opposite to each detected edge.

30 63. The apparatus according to claim 62, wherein said target pixel is
determined to be a thickening pixel if said target pixel has two edges and if the
horizontal and vertical gradients of a pixel at the junction of said two edges and
diagonally abutting said target pixel are absolute local minima.

64. The apparatus according to claim 63, further comprising:

35 means for determining if pixels in a predetermined size neighbourhood of
said target pixel are also candidates for thickening; and

means for restoring said target pixel to its original value if said neighbouring
pixels are also thickening candidates.

65. The apparatus according to claim 60, further comprising:
means for determining a zone containing said neighbourhood of pixels; and
means for negating said predetermined gradients of said target and
neighbouring pixels, if said zone is an inverted zone having a dark background
colour and a bright foreground colour.

5 66. The apparatus according to claim 65, wherein said means for
determining a zone further comprises:
means for choosing a representative window for said target pixel;
means for comparing local statistics regarding luminance of pixels in the
10 representative window to obtain zone; and
means for smoothing the zone.

67. An apparatus for smoothing jagged edges in graphical data, said
apparatus comprising:

15 means for detecting a plurality of zones of said graphical data dependent
upon the values of a plurality of pixels contained in each zone, said zones each
containing a minority of either bright or dark pixels or containing an equal number of
bright and dark pixels; and

means for filtering pixels of each zone, said zone filtering means further
comprising:

20 means for antialiasing said zone if said bright and dark
pixels are substantially equally present;

means for thickening said zone if said dark pixels are in a
minority in said zone or if said bright pixels are in a minority in
said zone.

25 68. The apparatus according to claim 67, wherein said antialiasing means
further comprises:

means for determining, for each pixel of said zone processed as a target
pixel, one or more edges if the intensities of a target pixel and a neighbouring pixel,
both abutting a respective edge site, are substantially different, predetermined
30 gradients dependent upon said respective edge site are an absolute local maxima, and
predetermined gradients of diagonal pixels are not an absolute local maxima of the
same sign as that of the neighbouring pixel; and

means for modifying, for each pixel of said zone processed as a target pixel,
said target pixel using a filter having coefficient values dependent upon said one or
35 more determined edges.

69. The apparatus according to claim 67, wherein said thickening means
further comprises:

means for determining, for each pixel of said zone processed as a target pixel, one or more edges if the intensities of a target pixel and a neighbouring pixel, both abutting a respective edge site, are substantially different, and predetermined gradients dependent upon said respective edge site are an absolute local maxima; and
5 means for modifying, for each pixel of said zone processed as a target pixel, said target pixel using a filter having coefficient values dependent upon said one or more determined edges.

70. The apparatus according to claim 67, further comprising
10 means for applying antialiasing to said zone, if said dark or bright pixels are in the minority in said zone;

means for comparing said antialiased zone with said thickened zone; and
means for replacing values of one or more pixels in said thickened zone with antialiased values of a corresponding pixel in said antialiased zone dependent upon results of said comparison.

15 71. The apparatus according to claim 70, wherein:

said comparison means further comprises means for generating a difference map by subtracting values of pixels in said thickened zone from corresponding values in said antialiased zone, or vice versa; and

20 means for determining pixels of said thickened zone corresponding with non-zero values of said difference map and replacing said pixels of said thickened zone with corresponding values of said antialiased zone.

72. The apparatus according to claim 67, further comprising means adapted to repeatedly perform the operations of the detection and filtering means a predetermined number of iterations.

25 73. A computer program product including a computer readable medium having recorded thereon a computer program for smoothing jagged edges in graphical data, said computer program product comprising:

30 means for detecting one or more edges of a selected pixel in said graphical data dependent upon intensities of said selected pixel and another pixel surrounding a respective site of said one or more edges and predetermined gradients of at least said selected pixel and said respective site surrounding pixel; and

35 filtering means for applying an adapted convolution mask to said selected pixel and a predetermined neighbourhood of pixels containing said selected pixel, wherein coefficient values of said convolution mask are dependent upon said one or more detected edges.

74. The computer program product according to claim 73, wherein the detecting operation of the detection means is dependent upon a difference between the

intensities of said selected pixel and said site surrounding pixel satisfying an intensity threshold.

75. The computer program product according to claim 74, wherein the detecting operation of the detection means is dependent upon an absolute value of the difference exceeding said intensity threshold.

76. The computer program product according to claim 73, wherein the detecting operation of the detection means is dependent upon the predetermined gradients of said selected pixel and said site surrounding pixel satisfying a gradient threshold.

77. The computer program product according to claim 76, wherein the predetermined gradients are either vertical or horizontal dependent upon said edge site being either horizontal or vertical, respectively.

78. The computer program product according to claim 76, wherein the predetermined gradients are greater than said gradient threshold.

79. The computer program product according to claim 76, wherein the predetermined gradients are less than said gradient threshold.

80. The computer program product according to claim 76, wherein the detecting operation of the detection means is further dependent upon predetermined gradients of diagonal pixels in said neighbourhood satisfying a gradient threshold.

81. The computer program product according to claim 80, wherein:
the predetermined gradients of said selected pixel and said site surrounding pixel are greater than said gradient threshold; and
the predetermined gradients of said diagonal pixels are less than said gradient threshold.

82. The computer program product according to claim 80, wherein:
the predetermined gradients of said selected pixel and said site surrounding pixel are less than said gradient threshold; and
the predetermined gradients of said diagonal pixels are greater than said gradient threshold.

83. The computer program product according to claim 73, wherein if a single edge is detected, one or more coefficient values of a portion of said convolution mask corresponding to an edge containing said site surrounding pixel is set to a predetermined value.

84. The computer program product according to claim 83, wherein said predetermined value is zero.

85. The computer program product according to claim 73, wherein if a double edge is detected, one or more coefficient values of each portion of said

convolution mask corresponding to a respective edge containing one of the two site surrounding pixels are set to a predetermined value.

86. The computer program product according to claim 85, wherein said predetermined value is zero.

5 87. The computer program product according to claim 78 or 79, further comprising means for determining, for a detected double edge, whether said selected pixel is a thickening pixel dependent upon a horizontal gradient and a vertical gradient of a pixel at the junction of the two edges satisfy respective gradient thresholds.

88. The computer program product according to claim 87, wherein one or
10 more coefficient values of a portion of said convolution mask corresponding to each edge opposite to the detected double edges is set to a predetermined value.

89. The computer program product according to claim 73, wherein the
detecting operation of said detection means is carried out for each colour component
of said selected pixel, and said one or more detected edges are dependent upon the
15 largest intensity and gradient values of the colour components.

90. A computer program product including a computer readable medium
having recorded thereon a computer program for smoothing jagged edges in graphical
data, said computer program product comprising:

means for determining one or more edges if the intensities of a target pixel
20 and a neighbouring pixel, both abutting a respective edge site, are substantially
different, predetermined gradients dependent upon said respective edge site are an
absolute local maxima, and predetermined gradients of diagonal pixels are not an
absolute local maxima of the same sign as that of the neighbouring pixel; and

filtering means for modifying said target pixel, said filtering means having
25 coefficient values dependent upon said one or more determined edges.

91. The computer program product according to claim 90, further
comprising means for setting to zero one or more coefficient values of a portion of
said filter corresponding to each detected edge.

92. The computer program product according to claim 90, further
30 comprising:

means for comparing said modified value of said target pixel with the
original value of said target pixel.

means for restoring said target pixel to its original value dependent upon
number of modified values of said graphical data.

93. The computer program product according to claim 90, further
35 comprising:

means for determining the number of modified pixels in a predetermined
neighbourhood of pixels containing said target pixel; and

means for restoring one or more modified pixels in said neighbourhood to the original value of the respective pixel, if said number exceeds a threshold number of modified pixels.

94. The computer program product according to claim 90, further comprising:

means for determining if modified pixels in a predetermined neighbourhood of pixels containing said target pixel conform to a predetermined pattern of pixels; and

means for restoring any pixels determined not to conform to the predetermined pattern to the original value of the respective pixel.

95. The computer program product according to claim 90, further comprising:

means for determining the number of modified pixels in a predetermined neighbourhood of pixels containing said target pixel having increased values;

means for determining the number of modified pixels in a predetermined neighbourhood of pixels containing said target pixel having decreased values; and

means for restoring the original value of said target pixel if the number of increased-value modified pixels is not substantially the same as the number of decreased-value modified pixels.

96. A computer program product including a computer readable medium having recorded thereon a computer program for smoothing jagged edges in graphical data, said computer program product comprising:

means for determining one or more edges if the intensities of a target pixel and a neighbouring pixel, both abutting a respective edge site, are substantially different, and predetermined gradients dependent upon said respective edge site are an absolute local maxima; and

filter means for modifying said target pixel, said filter means having coefficient values dependent upon said one or more determined edges.

smoothing the zone.

97. The computer program product according to claim 96, further comprising means for setting to zero one or more coefficient values of a portion of said filter corresponding to each detected edge.

98. The computer program product according to claim 96, further comprising:

means for determining if a target pixel is a thickening pixel; and

means for setting to zero one or more coefficient values of a portion of said filter corresponding to an edge opposite to each detected edge.

99. The computer program product according to claim 98, wherein said target pixel is determined to be a thickening pixel if said target pixel has two edges and if the horizontal and vertical gradients of a pixel at the junction of said two edges and diagonally abutting said target pixel are absolute local minima.

5 100. The computer program product according to claim 99, further comprising:

means for determining if pixels in a predetermined size neighbourhood of said target pixel are also candidates for thickening; and

10 means for restoring said target pixel to its original value if said neighbouring pixels are also thickening candidates.

101. The computer program product according to claim 96, further comprising:

means for determining a zone containing said neighbourhood of pixels; and

15 means for negating said predetermined gradients of said target and neighbouring pixels, if said zone is an inverted zone having a dark background colour and a bright foreground colour.

102. The computer program product according to claim 101, wherein said means for determining a zone further comprises:

means for choosing a representative window for said target pixel;

20 means for comparing local statistics regarding luminance of pixels in the representative window to obtain zone; and

means for smoothing the zone.

103. A computer program product including a computer readable medium having recorded thereon a computer program for smoothing jagged edges in graphical data, said computer program product comprising:

25 means for detecting a plurality of zones of said graphical data dependent upon the values of a plurality of pixels contained in each zone, said zones each containing a minority of either bright or dark pixels or containing an equal number of bright and dark pixels; and

30 means for filtering pixels of each zone, said zone filtering means further comprising:

means for antialiasing said zone if said bright and dark pixels are substantially equally present;

35 means for thickening said zone if said dark pixels are in a minority in said zone or if said bright pixels are in a minority in said zone.

104. The computer program product according to claim 103, wherein said antialiasing means further comprises:

means for determining, for each pixel of said zone processed as a target pixel, one or more edges if the intensities of a target pixel and a neighbouring pixel, both abutting a respective edge site, are substantially different, predetermined gradients dependent upon said respective edge site are an absolute local maxima, and
5 predetermined gradients of diagonal pixels are not an absolute local maxima of the same sign as that of the neighbouring pixel; and

means for modifying, for each pixel of said zone processed as a target pixel, said target pixel using a filter having coefficient values dependent upon said one or more determined edges.

10 105. The computer program product according to claim 103, wherein said thickening means further comprises:

means for determining, for each pixel of said zone processed as a target pixel, one or more edges if the intensities of a target pixel and a neighbouring pixel, both abutting a respective edge site, are substantially different, and predetermined
15 gradients dependent upon said respective edge site are an absolute local maxima; and

means for modifying, for each pixel of said zone processed as a target pixel, said target pixel using a filter having coefficient values dependent upon said one or more determined edges.

20 106. The computer program product according to claim 103, further comprising

means for applying antialiasing to said zone, if said dark or bright pixels are in the minority in said zone;

means for comparing said antialiased zone with said thickened zone; and

25 means for replacing values of one or more pixels in said thickened zone with antialiased values of a corresponding pixel in said antialiased zone dependent upon results of said comparison.

107. The computer program product according to claim 106, wherein:

30 said comparison means further comprises means for generating a difference map by subtracting values of pixels in said thickened zone from corresponding values in said antialiased zone, or vice versa; and

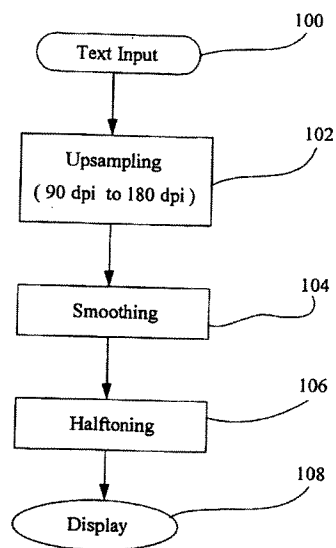
means for determining pixels of said thickened zone corresponding with non-zero values of said difference map and replacing said pixels of said thickened zone with corresponding values of said antialiased zone.

35 108. The computer program product according to claim 103, further comprising means adapted to repeatedly perform the operations of the detection and filtering means a predetermined number of iterations.

Dated 6 January, 1999
Canon Kabushiki Kaisha

Patent Attorneys for the Applicant/Nominated Person
SPRUSON & FERGUSON

2
2
2
2
2
2

**Fig. 1**

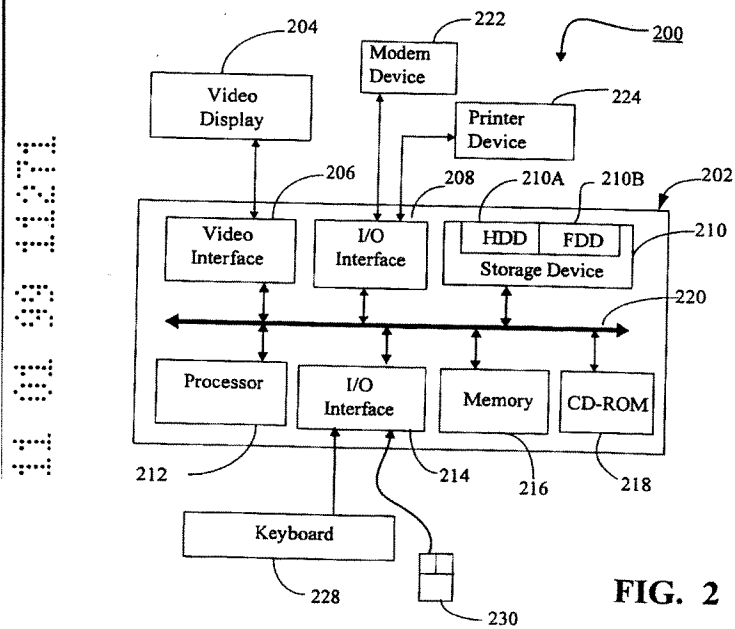


FIG. 2

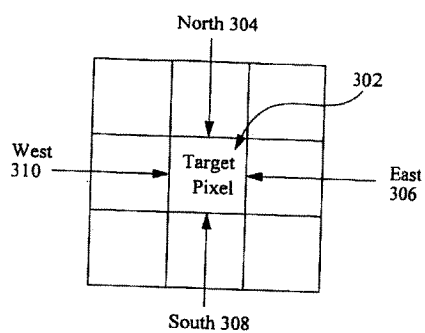


Fig. 3

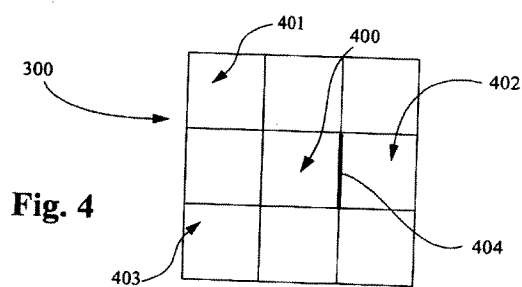


Fig. 4

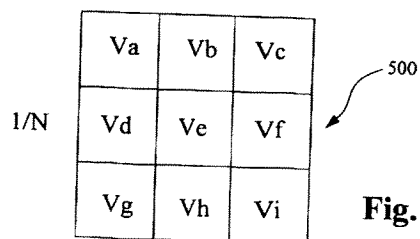


Fig. 5

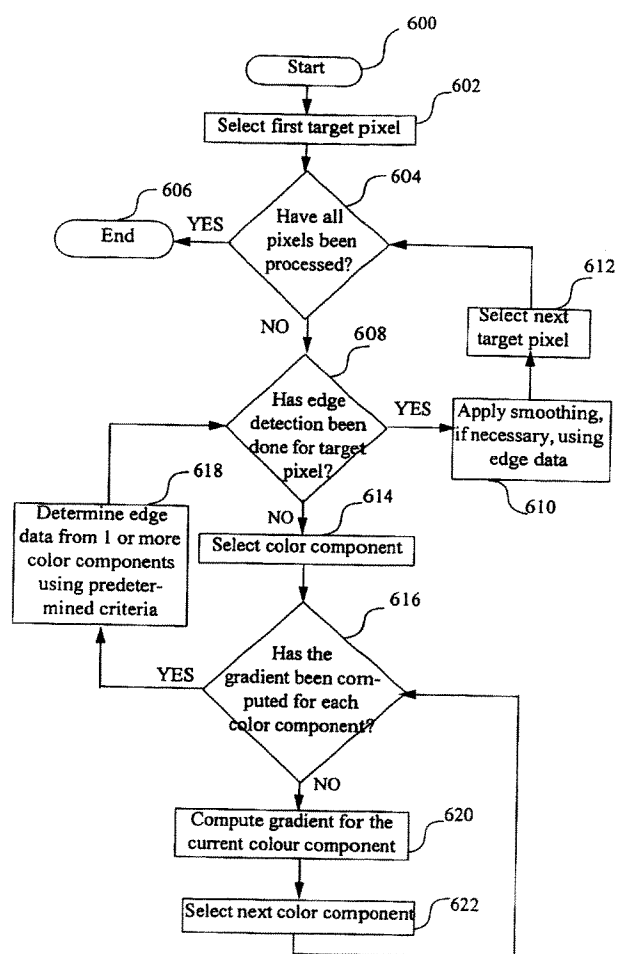


Fig. 6A

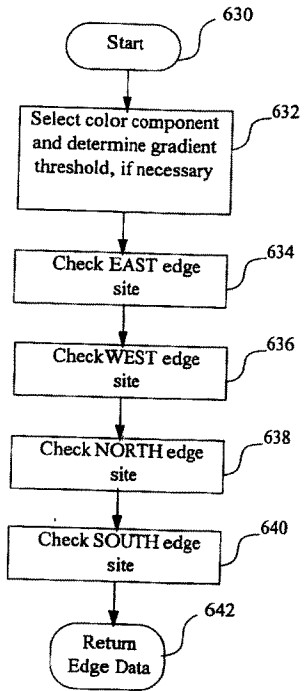
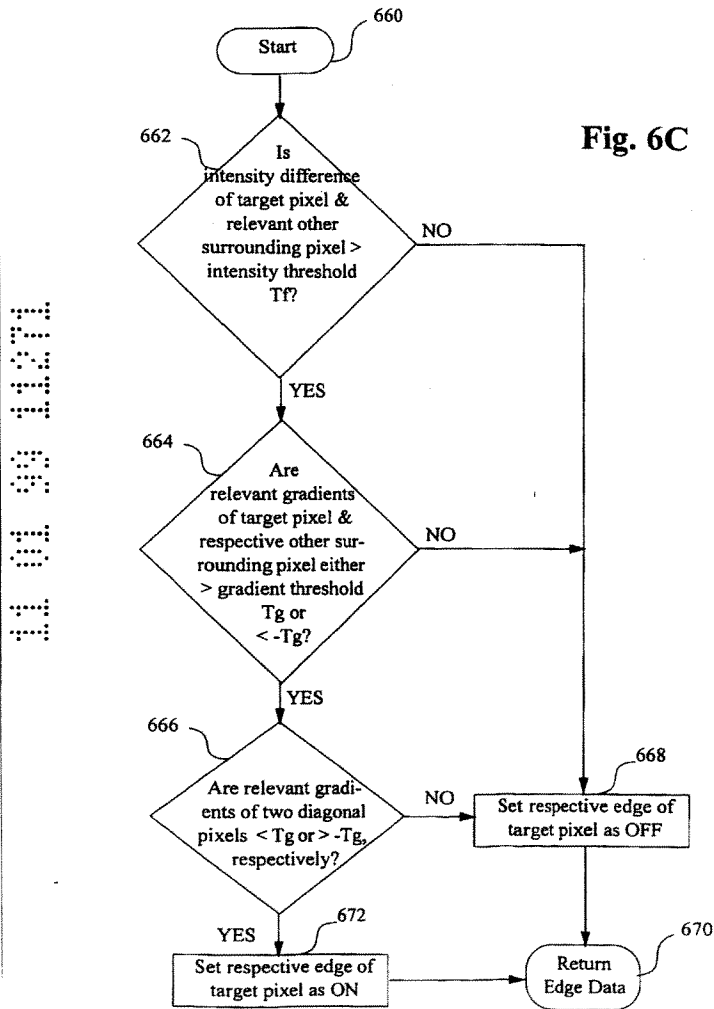
**Fig. 6B**

Fig. 6C



P1.CD20 385682 CFPunkin

[a:\csrc\filedraw\filed20\385682fig fra

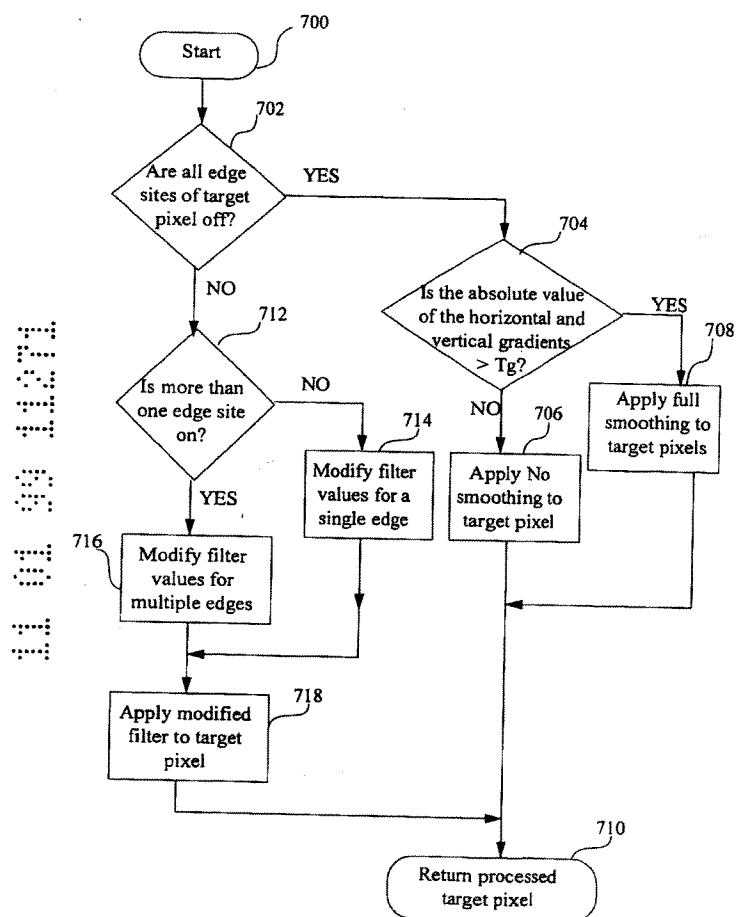
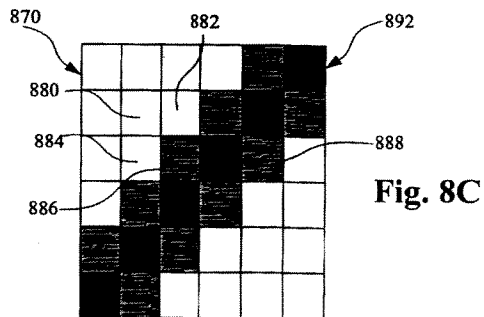
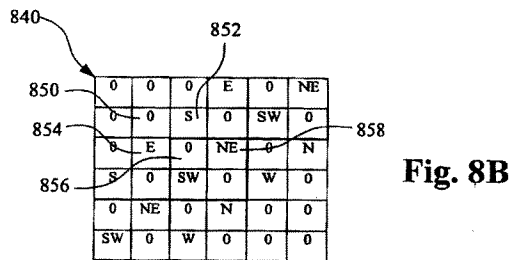
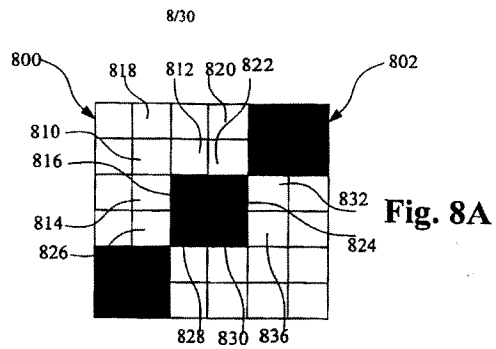


Fig. 7



9/30

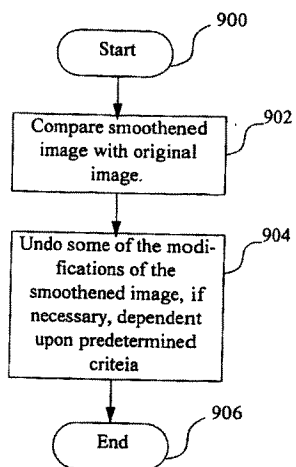


Fig. 9A

10/30

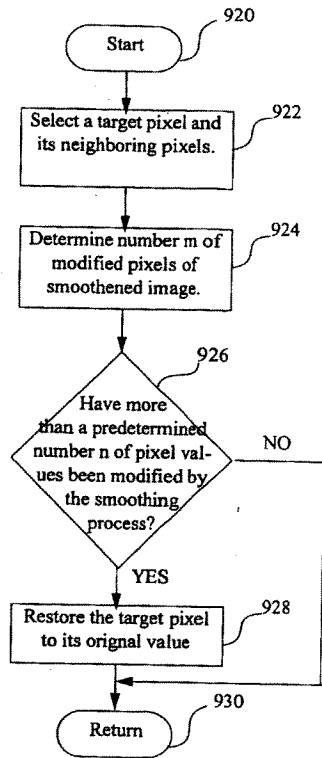
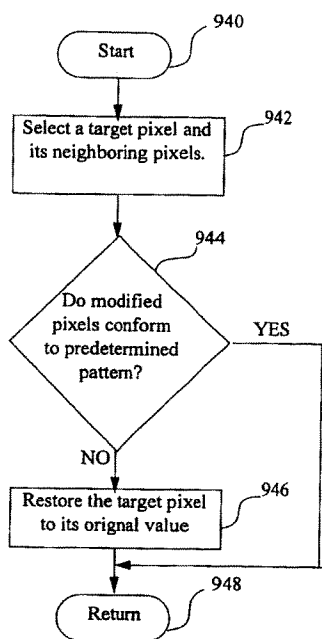


Fig. 9B

**Fig. 9C**

12/30

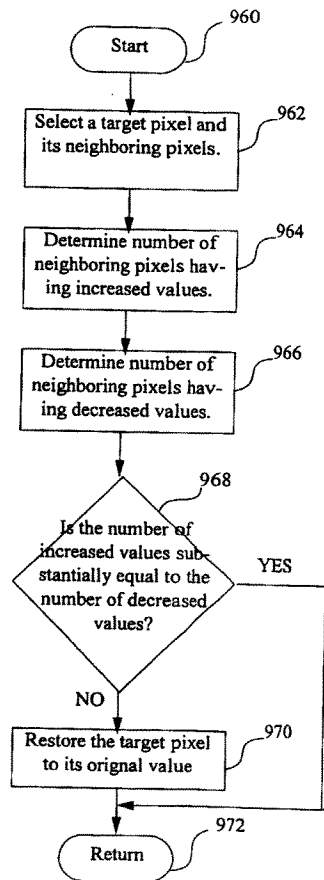


Fig. 9D

13/30

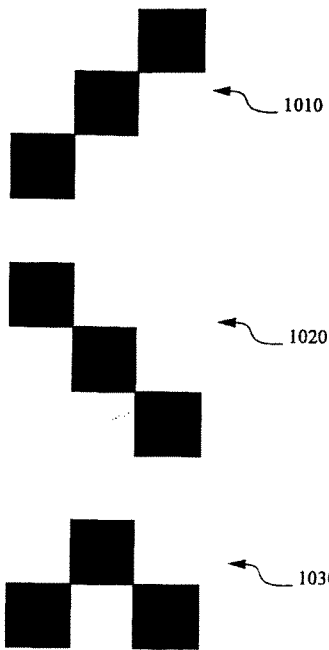
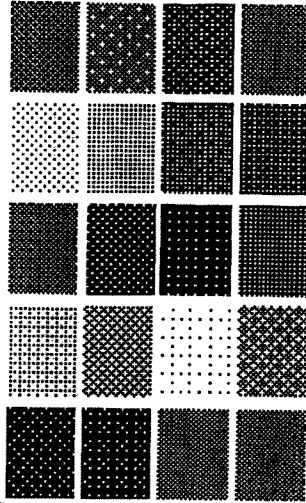


Fig. 10A

14/30



1050

Fig. 10B

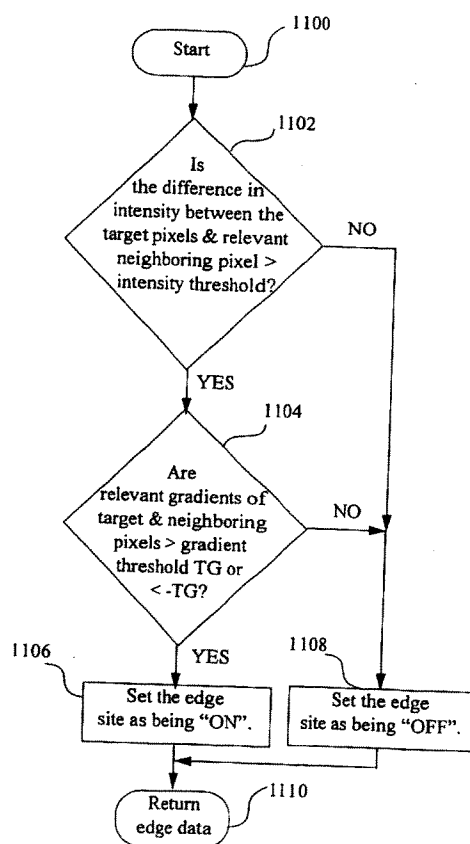


Fig. 11

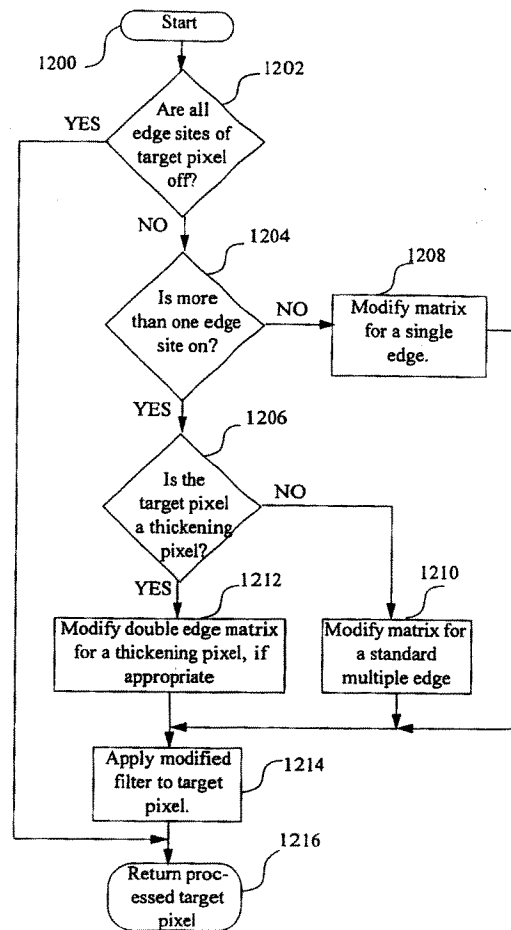
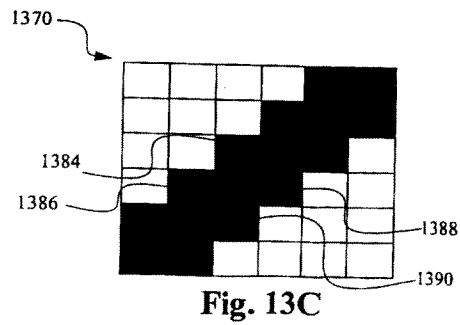
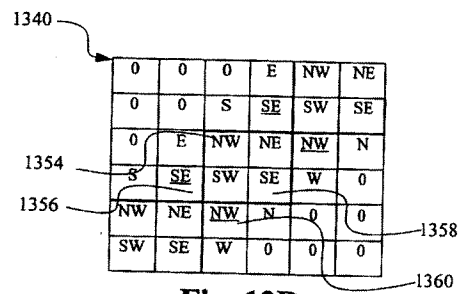
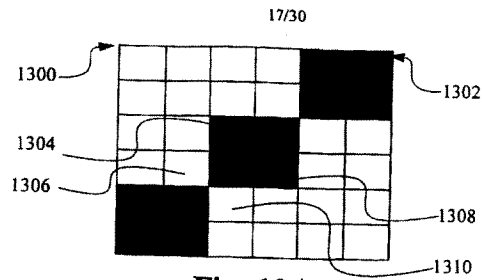


Fig. 12



1400
 ABCDEFGHIJKLMNOPQRSTUVWXYZ12345
 67890[]{}?/<>.,!@#\$%^&* _+=-\\'|~
 abcdefghijklmnopqrstuvwxyz12345|

Fig. 14A

1450
 ABCDEFGHIJKLMNOPQRSTUVWXYZ12345
 67890[]{}?/<>.,!@#\$%^&* _+=-\\'|~
 abcdefghijklmnopqrstuvwxyz12345|

Fig. 14B

1500
JUNETは全体的管理者（j
インの管理者の集合体（postmas
上位ドメインの管理者の両者に
て全員が責任を負っているわけ
が）の大学が運営しているとか書い
れは間違いです。前者は根も葉も
しかに一部の大学はJUNETの
ているといえるかもしれません

Fig. 15A

1502

1520
JUNETは全体的管理者（j
インの管理者の集合体（postmas
上位ドメインの管理者の両者に
て全員が責任を負っているわけ
が）の大学が運営しているとか書い
れは間違いです。前者は根も葉も
しかに一部の大学はJUNETの
ているといえるかもしれません

Fig. 15B

1522

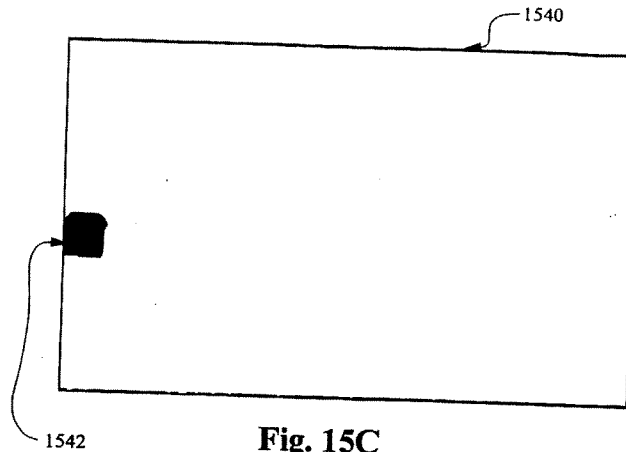
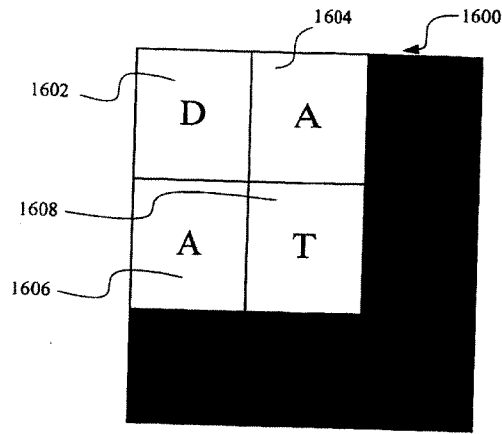
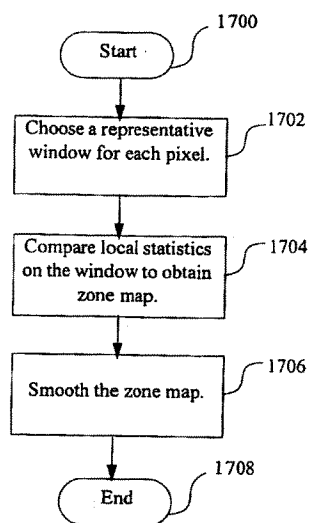


Fig. 15C

JUNETは全体的管理者（
 インの管理者の集合体（postmas
 上位ドメインの管理者の両者に
 て全員が責任を負っているわけ
 か）の大学が運営しているとか書い
 れは間違いです。前者は根も葉も
 しかに一部の大学はJUNETの
 ているといえるかもしれません

Fig. 15D

**Fig. 16**

**Fig. 17A**

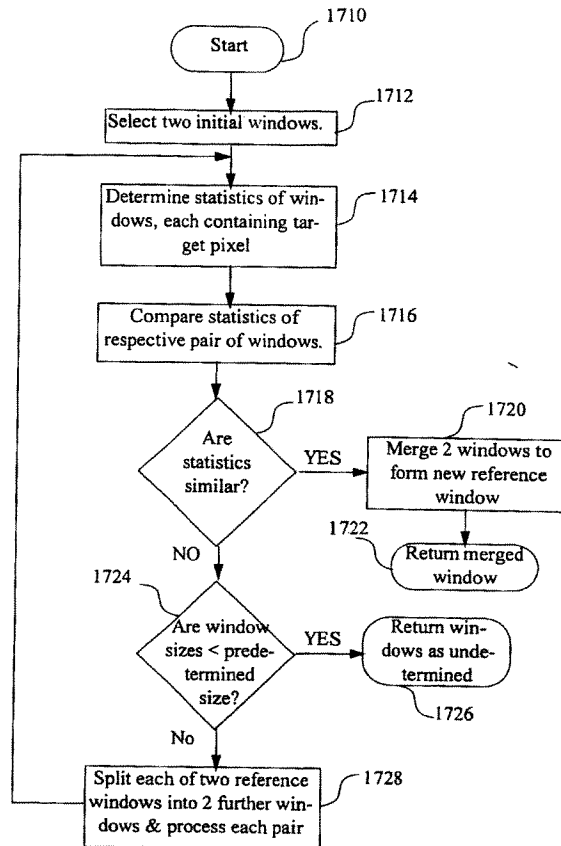


Fig. 17B

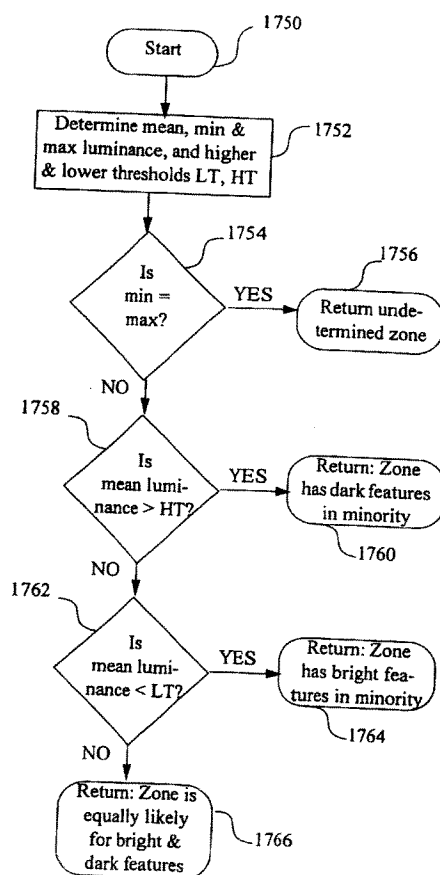


Fig. 17C

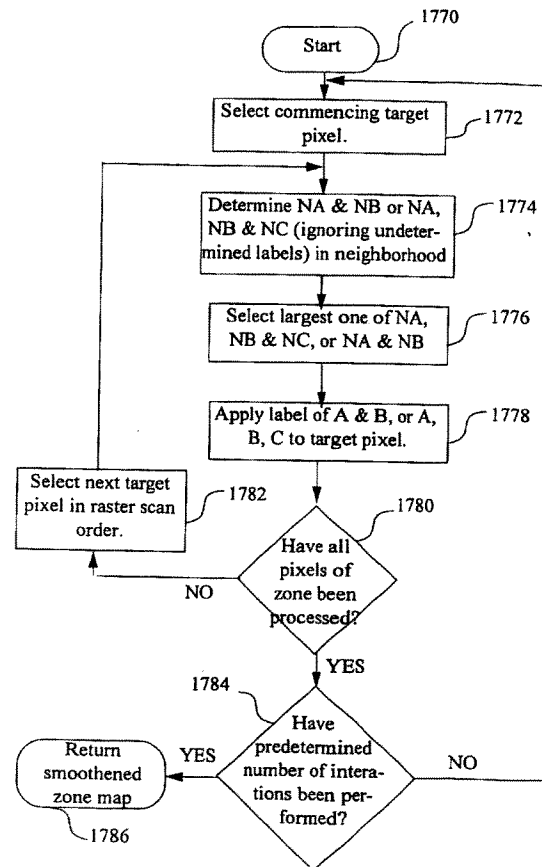
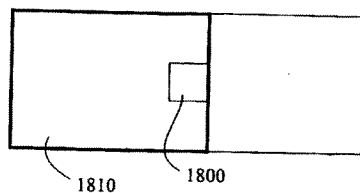
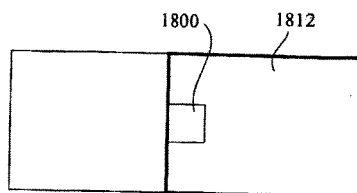
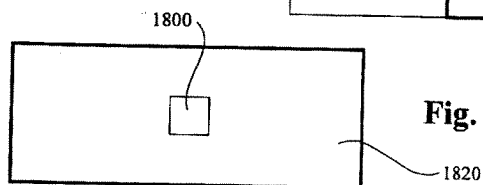
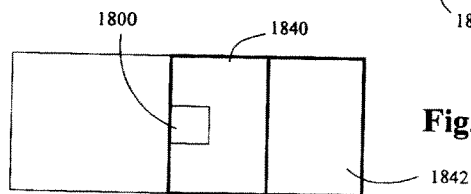
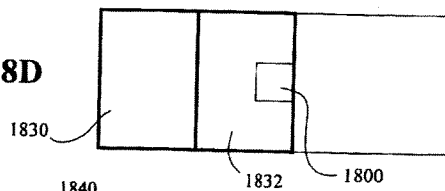


Fig. 17D

**Fig. 18A****Fig. 18B****Fig. 18C****Fig. 18D****Fig. 18E**

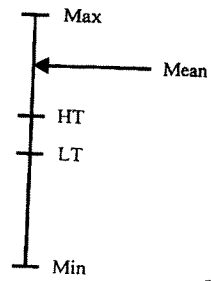


Fig. 19A

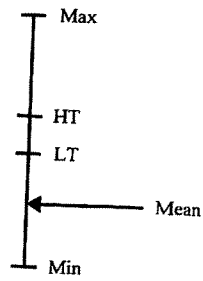


Fig. 19B

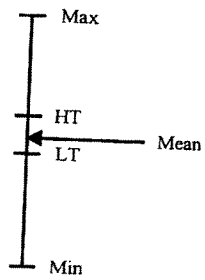
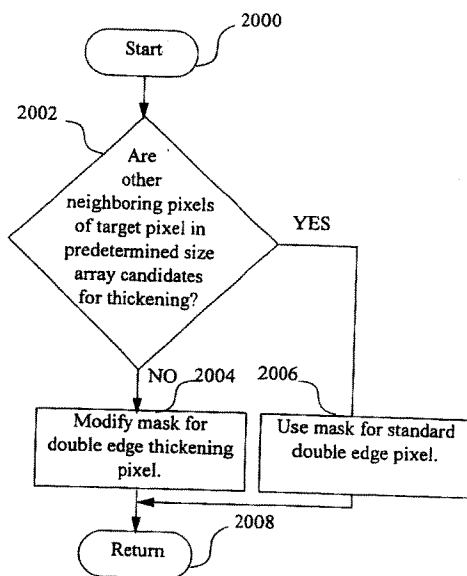


Fig. 19C

**Fig. 20**

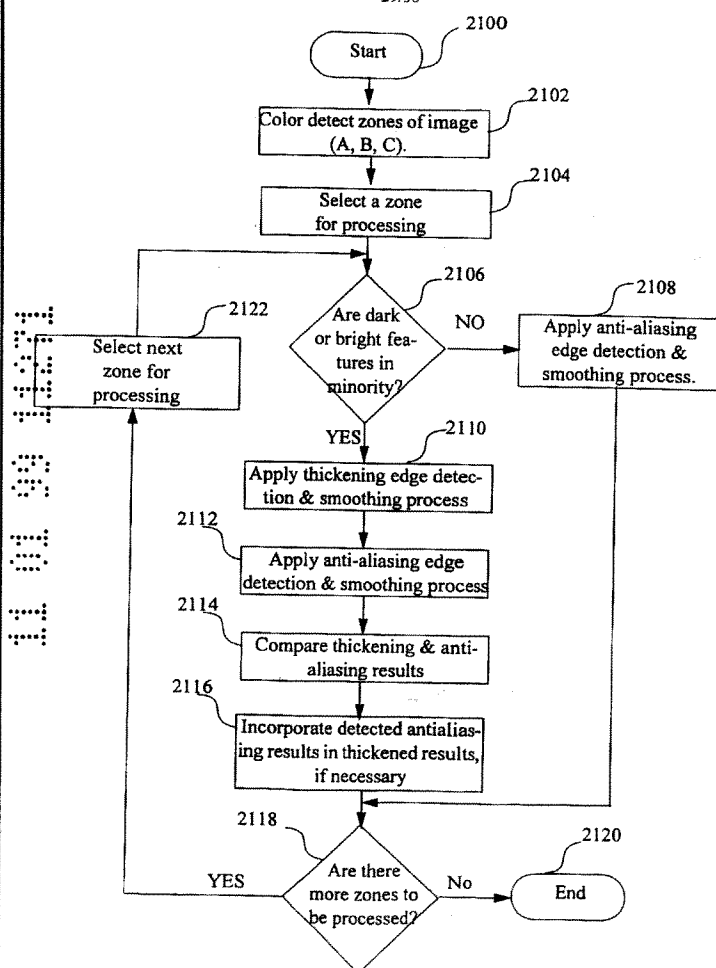


Fig. 21

FLCD20 385682 CFPunkn

[0:\cisra\filednew\filed20\385682\g.fra]

2200

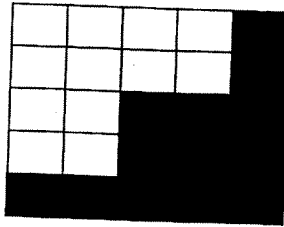


Fig. 22A

2210

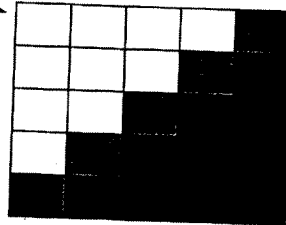


Fig. 22B

2220

0	0	0	0	+
0	0	0	-	0
0	0	+	0	0
0	-	0	0	0
+	0	0	0	0

2220

Fig. 22C